Efficient Linkable Ring Signature from Vector Commitment inexplicably named Multratug

Anton A. Sokolov

acmxddk@gmail.com

Abstract *In this paper we revise the ideas of our previous work 'Lin2-Xor lemma and Log-size Linkable Threshold Ring Signature' and introduce another lemma called Lin2-Choice, which extends the Lin2-Xor lemma. Using the Lin2-Choice lemma we create a compact general-purpose trusted-setup-free log-size linkable threshold ring signature with a strong security model. The signature size is* $2\log_2(n+1)+3l+1$*, where n is the ring size and l is the threshold. It is composed of several public coin arguments that are special honest verifier zero-knowledge and have computational witness-extended emulation. We use an arbitrary vector commitment argument as the base building block, providing the possibility to use any of its concrete implementations that have the above properties. Also, we present an extended version of our signature of size* $2\log_2(n+l+1)+6l+4$*, that simultaneously proves the sum of hidden amounts attached to the signing keys, i.e. proves the balance. All this in a prime order group without bilinear parings in which the decisional Diffie-Hellman assumption holds.*

Keywords: ring signature, linkable ring signature, log-size signature, threshold, anonymity, blockchain, hidden amounts, sum proof, zero-knowledge, unforgeability, non-frameability, witness-extended emulation.

1 INTRODUCTION

In the previous paper [11] we created a log-size linkable threshold ring signature based on the Lin2-Xor lemma, which we proved there. Now we want to know two things, namely, can we generalize the Lin2-Xor lemma using an arbitrary vector commitment argument that has computational witness-extended emulation (WEE) and is special honest verifier zero-knowledge (HVZK)? Also, can we get a linkable threshold ring signature out of it that is more efficient in size and verification time?

We answer both of these questions in the affirmative. Lin2-Choice lemma we present herein and its accompanying efficient ring signature seem to be useful findings. Our new ring signature keeps using the linking tag of the form $x^{-1}\mathcal{H}_{\mathbf{point}}(xG)$, and also has a version with the linking tag form $x\mathcal{H}_{\mathbf{point}}(xG)$, which is time-tested since the work by Liu, Wei, and Wong [7]. Although both these forms are indistinguishable from each other and from white noise, as we proved in [11].

The signature we present turns out to be extensible; we also introduce an extended version of it, which in addition to proving knowledge of signing keys also proves the sum of hidden amounts, i.e the balance. By proof of the sum of hidden amounts, the balance for short, we mean that prover demonstrates a blinded commitment to some secret amount and proves that this secret amount is equal to the sum of those amounts, which correspond to the actual signing keys and are also blinded. To construct the extended version of the signature we provide one more lemma, Lin2-2Choice, as we call it.

We will not repeat common words about signatures from the introduction of [11], they all remain valid. We will keep our presentation concise, taking into account that many explanations can be taken from [11] as well as from the work of Benedikt Bünz et al. [2]. As another basic ingredient, we will now use what we think is an elegant way of turning a protocol into zero-knowledge by adding noise in an orthogonal dimension to all transmitted elements, which we learned from the work of Heewon Chung et al. [3].

As for notation, we mainly use the notation from [11], supplementing it with notation from [2] and [3] where necessary, more on this in Section 2.1. Also, we use a kind of protocol representation inspired by [3].

Overall, in this paper we assume that a reader has an understanding of the works [2, 3, 11] and possesses an appropriate intuition, so we keep our descriptions and proofs brief, otherwise the paper would be too long. Moreover, since the methods of proving unforgeability, anonymity, and other protocol properties used in [2, 3] are already widely known, we describe only the key points for our proofs, believing that they suffice to reconstruct all the details of interest.

1.1 CONTRIBUTION

This work results in two state-of-the art trusted-setup-free pairings-free DDH-based log-size schemes. The first is called EFLRSL and is a linkable threshold ring signature, while the second is called Multratug and is a linkable threshold ring signature with balance proof. So the first one is a lightened version of the second. Their sizes and verification complexities are shown in Table 4. In addition, we provide a version of Multratug with a linear by private key linking tag, which is only l elements larger, this is reflected in Table 5.

Multratug can be used in blockchain. It integrates easily with the range proofs proposed in [2, 3], thus providing everything usually required for a typical transaction with hidden amounts. EFLRSL is general-purpose, it can be used in a wide range of trustless environments, especially where data size must be kept minimal. Both schemes support a strong security model defined by Theorems 9, 13. The model permits malformed and unevenly distributed keys in the rings, so the schemes are suitable for real environments that do not impose any restrictions on public keys, except that the latter themselves must not reveal their private keys.

A comparison with recently proposed solutions of the same class is represented in Tables 5, 6. It shows that EFLRSL and Multratug are at least on par with the most efficient ones known so far. For low threshold values, our proofs are shorter than most known solutions. However, as the threshold increases, they grow faster than the others. Nevertheless, we see that this is more than compensated by the fact that the verification complexities of EFLRSL and Multratug grow noticeably slower in this case. For the case of one actual signer, i.e., for threshold $l = 1$, both schemes are the shortest among those with strong security models.

Each of EFLRSL and Multratug is based on its own proof of membership. Both of the latter, in turn, are based on a plain vector commitment argument, for which we use variations of the log-size reduction in the spirit of inner product argument by Bünz et al. [2]. In the Lin2-Choice and Lin2-2Choice lemmas we prove the special honest verifier zero-knowledge (HVZK) and computational witness-extended emulation (WEE) properties for both of these membership proofs.

The Lin2-Choice lemma is a generalization of the Lin2-Xor lemma [11] to the case of n pairs of elements. Having a ring $\mathbf{P} = \{P_i\}_{i=0}^{n-1}$ of n elements and a commitment Z to an arbitrary element $P_s \in \mathbf{P}$, using the Lin2-Choice lemma it is possible to prove membership of Z in $\mathbf{P}$. We call this metaphorically selecting an element from $\mathbf{P}$. A novelty resides in the construction of this proof of membership, which in a nutshell looks as the following game. Although, we simplify it for this preview.

At start both prover and verifier have Z and $\mathbf{P}$. They jointly pick n helper generators $\mathbf{Q} = \{Q_i\}_{i=0}^{n-1}$ such that all elements of $\mathbf{P} \cup \mathbf{Q}$ are orthogonal to each other. The prover publishes an element F. Then the verifier releases challenges $\mathbf{c} = \{c_i\}_{i=0}^{n-1}$, and the prover replies with a scalar r. Next, the verifier releases random δ. Finally, the prover convinces the verifier using an arbitrary vector commitment argument that the element

$$Z + \delta r F$$

is a weighted sum, with weights known to the prover, of the elements from the set

$$\{P_i + \delta c_i Q_i\}_{i=0}^{n-1}.$$

Of course, the vector commitment argument is to be HVZK and has to have WEE. Also, note, the commitment Z and all elements published by prover are blinded, we omit showing the blinding components for simplicity.

It appears to be that the above game completes successfully only if there is some scalar p known to the prover such that $p^{-1}Z \in \mathbf{P}$. The Lin2-Choice lemma guarantees this. Moving on, adding to this proof of membership a linking tag of the form $x^{-1}\boldsymbol{\mathcal{H}}_{\mathbf{point}}(xG)$ and optimizing the involved vector commitment argument, we obtain the EFLRSL signature of size

$$2\log_2(n+1) + 3l + 1.$$

The optimized vector commitment argument is presented in Section 6.4; it imposes a requirement that $(n+1)$ be a power of 2.

Turning to the hidden amounts, we assume that the ring $\mathbf{P}$ is amended with the set $\mathbf{A} = \{A_i\}_{i=0}^{n-1}$ such that for each index i the key P_i corresponds to the hidden amount A_i. Also, we assume tha the total hidden amount $A^{\mathbf{sum}}$ is specified, and the balance with it needs to be proved. We might subtract $A^{\mathbf{sum}}$ from each A_i and prove that for the actual signer this difference contains only the blinding component, as it is done e.g. in [9], however this would prevent us from creating an effective threshold version. Therefore, we specify the set $\mathbf{A}^{\mathbf{tmp}} = \{A_k^{\mathbf{tmp}}\}_{k=0}^{l-1}$ of re-hidden amounts corresponding to the actual signers and, simply put, add them to the end of the ring.

So, the simplified game is that at start both prover and verifier have $Z, \mathbf{P}, \mathbf{A}, \mathbf{A}^{\mathbf{tmp}}, \mathbf{Q}$ such that $\mathbf{Q}$ is zoomed to $(n+l)$ generators and all elements of $\mathbf{P} \cup \mathbf{A} \cup \mathbf{A}^{\mathbf{tmp}} \cup \mathbf{Q}$ are orthogonal to each other. It is impossible to ensure the orthogonality of regular addresses and hidden amounts taken from a blockchain, however it is easily achieved by adding their hashes-to-curve, we omit showing them in this preview. For $k \in [0 \dots l-1]$, the prover publishes the

elements F, E, the verifier releases $\mathbf{c} = \{c_i\}_{i=0}^{n+l-1}$, the prover replies with r, the verifier releases random $\delta_1, \delta_2, \omega$, the prover convinces the verifier that the element

$$Z + \delta_1 r F + \delta_2 c_{n+k} E$$

is a weighted sum, with weights known to the prover, of the elements from the set

$$\{P_i - \omega A_i + \delta_1 c_i Q_i\}_{i=0}^{n-1} \cup \{\omega A_{i-n}^{\mathbf{tmp}} + \delta_2 c_i Q_i\}_{i=n}^{n+l-1}.$$

The Lin2-2Choice lemma guarantees this game completes successfully only if the prover knows scalar p and index s such that $p^{-1}Z = P_s \in \mathbf{P} \wedge A_s = A_k^{\mathbf{tmp}}$, of course, omitting blinding components everywhere in this preview. After that, it only remains to check $\sum_{k=0}^{l-1} A_k^{\mathbf{tmp}} = A^{\mathbf{sum}}$, and the Multratug signature with the balance proof is ready. Its size is

$$2\log_2(n+1) + 6l + 4.$$

Thus, the contribution includes not only the final signatures EFLRSL and Multratug, the two membership proofs and the corresponding lemmas can also be regarded as something new.

1.2 METHOD OVERVIEW

In this paper we construct a number of protocols, which we then use as building blocks for our signatures. For each of the protocols we are interested in three properties, namely, completeness, HVZK, and WEE.

Completeness is seen from the protocol listings, we do not dwell on it. The HVZK property requires building a simulator, yet luckily each our protocol has a property which simplifies things. Namely, besides the fact that all scalars of the protocol transcripts are masked with independent and uniformly sampled summands, each element of the transcripts, except for completely dependent elements, has the form

$$X + \mu H, \tag{1}$$

where X is a semantic component of the element, H is a blinding generator built in such a way as to be clearly orthogonal to everything else, and μ is always an independent uniformly sampled scalar. Therefore, we refer to the work [3], where the situation is the same and simulator is constructed. We imply that for each of our protocols a simulator is constructed in the same way.

For each protocol we prove the WEE property in detail by constructing an extractor that restores witness by performing polynomial number of rewindings. We also prove that the obtained witness meets the limits specified in protocol's relation, otherwise the extractor breaks the DL assumption in a polynomial number of steps.

Thus, by the above, all our signatures rely on a complete, HVZK, and WEE underlying proving systems. All additional signature elements, except for the linking tags also called as key images, have the form (1) and, thus, do not reveal any information. Therefore, to establish unforgeability, anonymity, and other properties of our signatures, we refer to the works [7, 4, 11] where the same properties are obtained by the same means using key images of the forms $x\boldsymbol{\mathcal{H}}_{\mathbf{point}}(xG)$ and $x^{-1}\boldsymbol{\mathcal{H}}_{\mathbf{point}}(xG)$, which are proven indistinguishable from each other in [11].

As an optimization that saves two elements of the transcript space, we compress a prover's reply of $n+1$ scalars in one of our HVZK and WEE protocols by transmitting only a proof of knowledge of these scalars, not the scalars themselves. Since each of the scalars looks like an independent uniform randomness, we use a proof of their knowledge which is not HVZK per se, however, this does not override the HVZK property of the enclosing protocol, nor does it override the ability to create a simulated transcript indistinguishable from a real one. And, in sum, this does not revoke the possibility of referring to the same methods for proving the properties of the resulting signatures.

1.2.1 TWO ELEMENT COMMITMENT

The first helper sub-protocol is a two-element commitment argument. We denote it as

$$\texttt{zk2ElemComm}(X, H, Y;\, x, h).$$

In this notation, the elements X, H, Y are common input for prover and verifier, and x, h are prover's private input, that is, they are witnesses known only for it. The $\texttt{zk2ElemComm}(X, H, Y;\, x, h)$ argument proves the relation

$$\mathcal{R} = \{\, X, H \in \mathbb{G}^*, Y \in \mathbb{G};\, x, h \in \mathbb{F}_{\mathfrak{p}} \mid Y = xX + hH \,\}, \tag{2}$$

where X and H are orthogonal to each other. Also, we require the argument to be HVZK and WEE. In order to rely on something concrete in calculating the size and complexity of our next protocols, in Figure 2 we provide an uncomplicated implementation for it.

In sum, $\texttt{zk2ElemComm}(X, H, Y; x, h)$ convinces verifier that prover knows a representation of element Y as a weighted sum of orthogonal generators X and H with weights known to prover. We use a two-generators extension of the Schnorr identification scheme as an implementation of this proof. Its size is one element in $\mathbb{G}$ and two scalars in $\mathbb{F}_{\bar{\mathsf{p}}}$.

1.2.2 VECTOR COMMITMENT

Vector commitment argument

$$\texttt{zkVC}_n(\mathbf{X}, H, Y; \mathbf{a}, \alpha)$$

provides a proof for the relation

$$\mathcal{R} = \{\, \mathbf{X} \in \mathbb{G}^{n*}, H \in \mathbb{G}^*, Y \in \mathbb{G};\ \mathbf{a} \in \mathbb{F}_{\bar{\mathsf{p}}}^n, \alpha \in \mathbb{F}_{\bar{\mathsf{p}}} \mid Y = \langle \mathbf{a}, \mathbf{X} \rangle + \alpha H \,\}, \tag{3}$$

where all generators from the set $\mathbf{X} \cup \{H\}$ are orthogonal to each other. That is, $\texttt{zkVC}_n$ convinces verifier that prover knows $n+1$ weights, namely, $\mathbf{a}$ and α, in the decomposition of Y by the generators $\mathbf{X} \cup \{H\}$. The genearator H together with its corresponding weight α is used here to turn the protocol into zero-knowledge, as in [3].

Our implementation of $\texttt{zkVC}_n$ in Figure 3 is based on the inner product argument implementation from [2] for the relation

$$\mathcal{R} = \{\, \mathbf{G}, \mathbf{H} \in \mathbb{G}^{n*}, U, P \in \mathbb{G};\ \mathbf{a}, \mathbf{b} \in \mathbb{F}_{\bar{\mathsf{p}}}^n \mid P = \langle \mathbf{a}, \mathbf{G} \rangle + \langle \mathbf{b}, \mathbf{H} \rangle + \langle \mathbf{a}, \mathbf{b} \rangle\, U \,\}, \tag{4}$$

which we modify as follows. First, since we don't really need the inner product argument, just only its vector commitment part, we zero out the vector $\mathbf{b}$ in the relation (4), making the inner product $\langle \mathbf{a}, \mathbf{b} \rangle$ equal to zero everywhere and leave only the vector commitment, i.e. only the argument for the relation

$$\mathcal{R} = \{\, \mathbf{G} \in \mathbb{G}^{n*},\ P \in \mathbb{G};\ \mathbf{a} \in \mathbb{F}_{\bar{\mathsf{p}}}^n \mid P = \langle \mathbf{a}, \mathbf{G} \rangle \,\}. \tag{5}$$

Second, we add zero-knowledge property to the inner product argument not the way it is done in [2], instead we add it in a straighter way, as in [3]. That is, we respectively add the blinding summands αH, βH, and γH to the vector commitment P and to the L and R elements that are transmitted in the argument implementation in [2]. The secret factors α, β, γ are uniformly sampled from $\mathbb{F}_{\bar{\mathsf{p}}}^*$, the generator H is chosen independently, and thus P and all the transmitted L's and R's appear indistinguishable from random noise. We rename the vector $\mathbf{G}$ and the commitment P in the relation (5) as $\mathbf{X}$ and Y in the relation (3), respectively. The blinding summand αH is taken into account in the relation (3).

Third, for the case $n = 1$ we use our own Schnorr-like HVZK and WEE protocol, which is different from sub-protocols used in [2] and [3]. Namely, we use $\texttt{zk2ElemComm}$ for the case, and this does not alter the properties of the entire $\texttt{zkVC}_n$ protocol. In any case, any HVZK and WEE protocol that proves $Y = \mathrm{lin}(X_0, H)$ will do instead of $\texttt{zk2ElemComm}$ for $n = 1$ in $\texttt{zkVC}_n$.

Thus, our $\texttt{zkVC}_n$ implementation of the vector commitment argument in Figure 3 has the same properties as the implementation of the inner product argument from [2] with $\mathbf{b} = \mathbf{0}^n$, plus it is HVZK and, of course, it remains to be having WEE.

If we compare our $\texttt{zkVC}_n$ protocol with the weighted inner product argument from [3], which is also based on the inner product argument from [2], then just as with the comparison against the inner product argument from [2] we zero out the vector $\mathbf{b}$, thus making the weighted inner product $\mathbf{a} \odot_y \mathbf{b}$ equal to zero. In doing so, we assume the weight y equal to 1 everywhere, and also use $\texttt{zk2ElemComm}$ for the case $n = 1$.

Note, actually our implementation of $\texttt{zkVC}_n$ is not based on the weighted inner product argument of [3], since we use neither 'weighted' in the sense of [3] nor 'inner product'. From [3] we only use the way we turn the argument into zero-knowledge, type of notation that we find concise and convenient, and also we borrowed from [3] the idea of using a custom Schnorr-like protocol for $n = 1$.

Overall, size of our zero-knowledge vector commitment argument $\texttt{zkVC}_n$ is $2\log_2(n) + 1$ elements from $\mathbb{G}$ and 2 scalar from $\mathbb{F}_{\bar{\mathsf{p}}}$. Here and elsewhere, when we use this implementation of $\texttt{zkVC}_n$ we consider n is a power of 2. Although, as we have already noted, we are not generally bound to a particular realization of $\texttt{zkVC}_n$, hence when we use an optimized implementation of it, such as the one defined in Section 6.4, this requirement for n changes.

1.2.3 RANDOM WEIGHTING FOR 3-TUPLES

Another auxiliary argument,

$$\texttt{zk3ElemRW}(P, Q, R, H, Z, F, E; a, \alpha, \beta, \gamma)$$

shown in Figure 4, connects a triplet of orthogonal elements (P, Q, R) with a triplet of arbitrary elements (Z, F, E). One of the two elements Q and R in the first triplet can be zero, in which case the other two elements of the triplet (P, Q, R) must be orthogonal to each other. So, the protocol `zk3ElemRW` proves the following relation

$$\mathcal{R} = \left\{ \begin{array}{l} P \in \mathbb{G}^*,\ Q, R \in \mathbb{G},\ H \in \mathbb{G}^*,\ Z, F, E \in \mathbb{G}; \\ a, \alpha, \beta, \gamma \in \mathbb{F}_{\bar{\mathsf{p}}} \end{array} \middle| \begin{array}{l} Z = aP + \alpha H \ \wedge \\ F = aQ + \beta H \ \wedge \\ E = aR + \gamma H \end{array} \right\}, \tag{6}$$

where it is required that all non-zero elements from the set $\{P, Q, R, H\}$ are orthogonal to each other, which is denoted as $\mathrm{ort}(\mathrm{nz}(P, Q, R, H))$, and that at least one of Q and R is non-zero, which can be written as $(Q + R) \in \mathbb{G}^*$.

There are two sampled challenges δ_1 and δ_2 within the protocol `zk3ElemRW`. The two sums X and Y together with total blinding factor $\hat{\alpha}$ are constructed via these challenges

$$\begin{aligned} X &= P + \delta_1 Q + \delta_2 R, \\ Y &= Z + \delta_1 F + \delta_2 E, \\ \hat{\alpha} &= \alpha + \delta_1 \beta + \delta_2 \gamma\,. \end{aligned}$$

As the second step, using an arbitrary complete, HVZK, and WEE argument it is proved that Y is a weighted sum of X and H with some known to prover weights. Thus the relation (6) is proven.

In terms of [11], in the second step of `zk3ElemRW` a proof of $Y = \mathrm{lin}(X, H)$ for prover is somehow obtained (in an HVZK and WEE way). We will be often omitting everything connected with H as a technical blinding detail, so writting down this shortly as $Y \sim X$ (to the accuracy of H).

Computational witness-extended emulation of the protocol `zk3ElemRW` can be proved by well-known methods, such as, e.g., in the RandomWeighting-WEE lemma proof in [11]. The extreme case, when one of the elements Q or R is zero, is not problematic.

1.2.4 SIMMETRIC VECTOR COMMITMENT

We will also need an argument to convince verifier that several, e.g. two or three, vector commitments share, except for blinding summands, the same coefficients known to prover. That is, we will need an argument

$$\mathtt{zkSVC}_{3,n}(\mathbf{P}, \mathbf{Q}, \mathbf{R}, H, Z, F, E;\ \mathbf{a}, \alpha, \beta, \gamma)$$

for the following relation

$$\mathcal{R} = \left\{ \begin{array}{l} \mathbf{P} \in \mathbb{G}^{n*}, \mathbf{Q}, \mathbf{R} \in \mathbb{G}^n, H \in \mathbb{G}^*, Z, F, E \in \mathbb{G}; \\ \mathbf{a} \in \mathbb{F}_{\bar{\mathsf{p}}}^n, \alpha, \beta, \gamma \in \mathbb{F}_{\bar{\mathsf{p}}} \end{array} \middle| \begin{array}{l} Z = \langle \mathbf{a}, \mathbf{P} \rangle + \alpha H \ \wedge \\ F = \langle \mathbf{a}, \mathbf{Q} \rangle + \beta H \ \wedge \\ E = \langle \mathbf{a}, \mathbf{R} \rangle + \gamma H \end{array} \right\}, \tag{7}$$

where all non-zero elements from the set $\mathbf{P} \cup \mathbf{Q} \cup \mathbf{R} \cup \{H\}$ are orthogonal to each other, written as

$$\mathrm{ort}(\mathbf{P} \cup \mathrm{nz}(\mathbf{Q}) \cup \mathrm{nz}(\mathbf{R}) \cup \{H\}),$$

and where for any index $i \in [0 \ldots n-1]$ at least one of two elements $\mathbf{Q}_{[i]}$ and $\mathbf{R}_{[i]}$ is nonzero, denoted as

$$(\mathbf{Q} + \mathbf{R}) \in \mathbb{G}^*.$$

The relation (7) states that three different vector commitments Z, F, E are sort of 'symmetrical' to each other by their common weights $\mathbf{a}$, which are applied to the bases $\mathbf{P}, \mathbf{Q}, \mathbf{R}$, respectively. The protocol $\mathtt{zkSVC}_{3,n}$ is shown in Figure 5.

Note again, that we require all elements in $\mathbf{P}$ to be non-zero, while vectors $\mathbf{Q}$ and $\mathbf{R}$ can contain zero elements, as long as for each index there is at least one non-zero element at that index in them. This condition is necessary for the protocol $\mathtt{zkSVC}_{3,n}$ to be implementable.

Using random weighting we reduce the argument $\mathtt{zkSVC}_{3,n}$ to the vector commitment argument $\mathtt{zkVC}_n$ at zero cost. Namely, for random δ_1 and δ_2 we construct

$$\begin{aligned} \mathbf{X} &= \mathbf{P} + \delta_1 \mathbf{Q} + \delta_2 \mathbf{R}, \\ Y &= Z + \delta_1 F + \delta_2 E, \\ \hat{\alpha} &= \alpha + \delta_1 \beta + \delta_2 \gamma\,, \end{aligned}$$

and call

$$\mathtt{zkVC}_n(\mathbf{X}, H, Y;\ \mathbf{a}, \hat{\alpha}).$$

As a result, we see that for each index $i \in [0 \dots n-1]$ the zk3ElemRW protocol is met, that means the relation (6) is fulfilled for each triplet pair (P_i, Q_i, R_i) and $(Z_{P_i}, F_{Q_i}, E_{R_i})$, and therefore the relation (7) is fulfilled. Here Z_{P_i} means P_i's component in decomposition of Z by the base $\mathbf{P}$, the same applies to F_{Q_i}, E_{R_i}. Of course, when the protocol $\mathtt{zkSVC}_{3,n}$ sucessfully completes, verifier is also convinced that the elements Z, F, E are weighted direct sums with weights known to prover of the vectors $\mathbf{P}, \mathbf{Q}, \mathbf{R}$, respectively.

1.2.5 LIN2-CHOICE LEMMA

In [11] we proved the Lin2-Xor lemma which, informally, allows us to select one pair of elements from two pairs of elements, i.e., it provides an argument for the relation

$$\mathcal{R} = \left\{ \ \mathbf{P}, \mathbf{Q} \in \mathbb{G}^{2*}, Z \in \mathbb{G}^*; s \in [0 \dots 1], \ p, q \in \mathbb{F}_{\bar{\mathfrak{p}}} \ \middle| \ \ Z = pP_s + qQ_s \ \right\}, \tag{8}$$

where the generators of $\mathbf{P} \cup \mathbf{Q}$ are orthogonal to each other. Also, in [11] by successive application of the Lin2-Xor lemma $\log_2(n)$ times we proved the Lin2-Selector lemma, which allows us to select one pair of elements from n pairs of elements. In other words, the Lin2-Selector lemma [11] provides an argument for the relation

$$\mathcal{R} = \left\{ \ \mathbf{P}, \mathbf{Q} \in \mathbb{G}^{n*}, Z \in \mathbb{G}^*; s \in [0 \dots n-1], \ p, q \in \mathbb{F}_{\bar{\mathfrak{p}}} \ \middle| \ \ Z = pP_s + qQ_s \ \right\}. \tag{9}$$

However, after some thought, we concluded that instead of proving the relation (9) by the Lin2-Selector lemma protocol, it is better to prove it directly as if the Lin2-Xor lemma applies to n pairs of element at once while making an auxiliary call to some vector commitment argument. This way is more efficient in size, and also gives more opportunities to optimize the verification complexity.

Intuition here is that in the first round of the Lin2-Xor lemma protocol both prover and verifier multiply one element in each of the two original pairs (P_0, Q_0) and (P_1, Q_1) by a random challenge, so that each of the two original pairs becomes a compound element with its random 'rotation', namely, they become $P_0 + c_0 Q_0$ and $P_1 + c_1 Q_1$. Here we use the notation and indexing from [11]. In the second round of the Lin2-Xor protocol, prover and verifier play a sub-protocol convincing the verifier that the element $Z + r_1 H_1$ is a linear combination of the two compound elements, which carry their random 'rotations' c_0 and c_1. It then turns out that this linear combination can be only one-hot, otherwise the DL assumption would be broken. Indeed, since $P_0, Q_0, P_1, Q_1, Z, H_1$ are fixed from the beginning, and as they are orthogonal to each other, the element $Z + r_1 H_1$ has at most one 'degree of freedom' parameterized by r_1. At the same time, each of the elements $P_0 + c_0 Q_0$ and $P_1 + c_1 Q_1$ has exactly one degree of freedom defined by the parameters c_0 and c_1 respectively. Hence, if both coefficients a, b in the linear combination

$$Z + r_1 H_1 = a(P_0 + c_0 Q_0) + b(P_1 + c_1 Q_1) \tag{10}$$

are not equal to zero, then the right-hand side of the equality (10), which has two 'degrees of freedom' with the random parameters c_0 and c_1, is balanced by one 'degree of freedom' of the left-hand side with the controlled parameter r_1, which is impossible without breaking orthogonality of P_0, Q_0, P_1, Q_1.

In line with this intuition, if we take n pairs of elements and turn them into n compound elements with random 'rotations' in the first round, and in the second round prove that $Z + r_1 H_1$ is a linear combination of these n compound elements, then exactly the same way we obtain that the compound element $Z + r_1 H_1$ with one 'degree of freedom' r_1 must balance the weighted sum of the compound elements of the form $P_i + c_i Q_i$, each adding one 'degree of freedom' to the right side of the equality

$$Z + r_1 H_1 = \sum_{i=0}^{n-1} a_i (P_i + c_i Q_i), \tag{11}$$

which is possible only if the vector of coefficients $\{a_i\}_{i=0}^{n-1}$ is one-hot. Thus, we obtain an argument for the relation (9) as a two-round game, where in the first round r_1 is chosen in response to n challenges $\{c_i\}_{i=0}^{n-1}$, and in the second round

$$\mathtt{zkVC}_n(\{P_i + c_i Q_i\}_{i=0}^{n-1}, H, Z + r_1 H_1 \,;\, \mathbf{a}, \alpha),$$

is played. Here H_1 is fixed as in [11], H is an independent generator for blinding, α is the blinding factor, and $\mathbf{a}$ is one-hot.

Also, since the vector $\mathbf{Q}$ carries only a technical role in the relation (9), in particular in [11] we get rid of Q_s by adding a proof that $q = 0$ everywhere in the signatures, we now include a proof of $q = 0$ in our argument. Taking everything into account, in the Lin2-Choice lemma (Theorem 5) we provide a HVZK and WEE protocol

$$\mathtt{zkLin2Choice}_n(\mathbf{P}, \mathbf{Q}, H, Z; s, p, \alpha)$$

shown in Figure 7 for the following relation

$$\mathcal{R} = \left\{ \begin{array}{l} \mathbf{P}, \mathbf{Q} \in \mathbb{G}^{n*}, H \in \mathbb{G}^*, Z \in \mathbb{G}; \\ s \in [0 \ldots n-1], p, \alpha \in \mathbb{F}_{\bar{\mathsf{p}}} \end{array} \;\middle|\; Z = pP_s + \alpha H \right\}, \tag{12}$$

where $\mathbf{P}, \mathbf{Q}, H$ satisfy ort($\mathbf{P} \cup \mathbf{Q} \cup \{H\}$) . Thus, our Lin2-Choice lemma allows to choose exactly one element from the set of orthogonal elements $\mathbf{P} \in \mathbb{G}^{n*}$.

Addressing the details, with the simultaneous proof of $q = 0$, the Lin2-Choice lemma protocol $\texttt{zkLin2Choice}_n$ for the relation (12) is constructed as follows

- After the first $\mathcal{P}$'s message both $\mathcal{P}$ and $\mathcal{V}$ have elements Z and F, where F plays the same role as H_1 in [11].
- All n elements of $\mathbf{Q}$ are multiplied by the challenges $\{c_i\}_{i=0}^{n-1}$ respectively, so $\mathcal{P}$ and $\mathcal{V}$ build a vector of elements $\hat{\mathbf{Q}} = \{c_i Q_i\}_{i=0}^{n-1}$.
- $\mathcal{P}$ replies with r, which plays the same role as r_1 in [11].
- $\mathcal{P}$ and $\mathcal{V}$ play $\texttt{zkSVC}_{2,n}(\mathbf{P}, \hat{\mathbf{Q}}, H, Z, rF;\ \mathbf{a}, \alpha, r\beta)$, where $\mathbf{a}$ is one-hot, H is an orthogonal blinding generator, α and β are blinding factors of Z and F respectively.

Informally, we can see that if $\mathbf{a}$ has more than one hot entry, then $\texttt{zkSVC}_{2,n}$ will not complete successfully for the same reason as the equality (11) will not hold for such $\mathbf{a}$. To be precise, the following equality is checked within $\texttt{zkSVC}_{2,n}$, and it guarantees $\mathbf{a}$ is one-hot

$$Z + \delta_1 rF = \sum_{i=0}^{n-1} a_i (P_i + \delta_1 c_i Q_i).$$

In addition to this, if $\texttt{zkSVC}_{2,n}$ completes successfully, then Z cannot contain elements from $\mathbf{Q}$ in the decomposition since $\texttt{zkSVC}_{2,n}$ guarantees $Z = \text{lin}(\mathbf{P} \cup \{H\})$.

1.2.6 SIGNATURE EFLRS1

Having zero-knowledge argument $\texttt{zkLin2Choice}_n$ for the relation (12), it is easy to build a ring signature, we call it EFLRS1 (Efficient linkable ring signature for 1 actual signer). Its interactive scheme is shown in Figure 10

$$\texttt{EFLRS1.SignAndVerify}_{1,n}(\mathsf{M}, \mathbf{P}; s, x).$$

By a ring we mean a set of n public keys

$$\mathbf{P} = \{P_i\}_{i=0}^{n-1}, \tag{13}$$

where $n \geqslant 1$. The signature convinces verifier that signer knows a scalar x such that the equality $P_s = xG$ holds for some $s \in [0 \ldots n-1]$. There are no assumptions about the public keys $\{P_i\}_{i=0}^{n-1}$, all they can be regarded as adversarially chosen.

By corresponding to the ring decoy set, technically called so, we will mean a set of n pairs of the form

$$\{\ (\ P_i + \zeta \boldsymbol{\mathcal{H}}_{\mathbf{point}}(P_i),\ Q_i\ )\ \}_{i=0}^{n-1}, \tag{14}$$

where P_i is a public key in the ring, ζ is a random weight, $\boldsymbol{\mathcal{H}}_{\mathbf{point}}$ is a hash to curve function, and $Q_i \in \mathbf{Q}$, where $\mathbf{Q}$ is a set of auxiliary orthogonal generators that can be prepared in advance, provided that $\boldsymbol{\mathcal{H}}_{\mathbf{point}}$ always generates elements orthogonal to $\mathbf{Q}$.

At the same time, key image is defined as

$$I = x^{-1} \boldsymbol{\mathcal{H}}_{\mathbf{point}}(P_s), \tag{15}$$

where x is a private key for the public key P_s such that there holds $P_s = xG$.

To obtain a signature it remains to define Z as

$$Z = G + \zeta I, \tag{16}$$

pick a blinding generator H as orthogonal to all other generators, and apply the protocol of the Lin2-Choice lemma as follows

$$\texttt{zkLin2Choice}_n(\{P_i + \zeta \boldsymbol{\mathcal{H}}_{\mathbf{point}}(P_i)\}_{i=0}^{n-1}, \mathbf{Q}, H, G + \zeta I; s, x^{-1}, 0),$$

thus producing the signature of size $2\log_2(n) + 6$.

When calculating the signature size we assume that the bitwise representation of an element from $\mathbb{G}$ takes as much space as the bitwise representation of a scalar from $\mathbb{F}_{\bar{\mathsf{p}}}$. We take into account all elements and scalars transmitted from prover to verifier, including the key image I. We ignore the ring of public keys $\{P_i\}_{i=0}^{n-1}$, which is assumed to be known beforehand for both prover and verifier.

Also, recalling that a signature signs an input message M for the first place, we use the well-known method of binding a signature to message, described, e.g. in [5]. Namely, we assume that the signature's random oracle depends of the input message, and thus the entire series of random values in each signature is bound to M.

1.2.7 MULTIPLE VECTOR COMMITMENTS

To create a threshold version of the signature we need one more helper zero-knowledge argument, namely, a proof of multiple vector commitment

$$\mathtt{zkMVC}_{l,n}(\mathbf{X}, H, \mathbf{Y};\, \mathfrak{a}, \boldsymbol{\alpha}),$$

that for a given element vector $\mathbf{Y} \in \mathbb{G}^l$ proves every $Y_i \in \mathbf{Y}$ is a vector commitment over the vector of orthogonal generators $\mathbf{X} \cup \{H\} \in \mathbb{G}^{n*} \times \mathbb{G}$, with the coefficients known to prover. It is shown in Figure 12, $\mathtt{zkMVC}_{l,n}$ is a protocol for the relation

$$\mathcal{R} = \{\, \mathbf{X} \in \mathbb{G}^{n*}, H \in \mathbb{G}^*, \mathbf{Y} \in \mathbb{G}^l;\, \mathfrak{a} \in \mathbb{F}_{\bar{\mathsf{p}}}^{l \times n}, \boldsymbol{\alpha} \in \mathbb{F}_{\bar{\mathsf{p}}}^{l} \mid \mathbf{Y} = \mathfrak{a} \cdot \mathbf{X} + \boldsymbol{\alpha} \cdot H \,\}\,. \tag{17}$$

The structure of this protocol is quite simple. All l elements from the vector $\mathbf{Y}$ are combined into one element Y with random weights, then the protocol $\mathtt{zkVC}_n$ proves that Y is a vector commitment over the generators $\mathbf{X} \cup \{H\}$, thus convincing verifier that, due to the random weights, every $Y_i \in \mathbf{Y}$ is a vector commitment over $\mathbf{X} \cup \{H\}$. This way we obtain a proof for a set of vector commitments at the price of one vector commitment proof.

1.2.8 MANY-OUT-OF-MANY PROOF

The $\mathtt{zkMVC}_{l,n}$ protocol, according to the relation (17), proves the same as l $\mathtt{zkVC}_n$ protocols prove. Now we will construct an efficient many-out-of-many proof of membership

$$\mathtt{zkLin2mChoice}_{n,l}(\mathbf{P}, \mathbf{Q}, H, \mathbf{Z}; \mathbf{s}, \mathbf{p}, \boldsymbol{\alpha}),$$

shown in Figure 13, for the following relation

$$\mathcal{R} = \left\{ \begin{array}{l} \mathbf{P}, \mathbf{Q} \in \mathbb{G}^{n*}, H \in \mathbb{G}^*, \mathbf{Z} \in \mathbb{G}^l; \\ \mathbf{s} \in [0 \ldots n-1]^l, \mathbf{p}, \boldsymbol{\alpha} \in \mathbb{F}_{\bar{\mathsf{p}}}^l \end{array} \;\middle|\; \begin{array}{l} \forall k \in [0 \ldots l-1] : \\ Z_k = p_k P_{s_k} + \alpha_k H \end{array} \right\}, \tag{18}$$

where $\mathbf{P}, \mathbf{Q}, H$ satisfy $\mathrm{ort}(\mathbf{P} \cup \mathbf{Q} \cup \{H\})$, which proves the same as l concurrent insances of the one-out-of-many proof of membership $\mathtt{zkLin2Choice}_n$ for the reletion (12), at the price of one instance.

All the l instances of $\mathtt{zkLin2Choice}_n$ are played as a sequence of nested sub-protocol calls invoked simultaneously, we depict this as follows

$$l \times \mathtt{zkLin2Choice}_n \hookrightarrow l \times \mathtt{zkSVC}_{2,n} \hookrightarrow\ l \times \mathtt{zkVC}_n\,. \tag{19}$$

Since each of these l concurrent $\mathtt{zkLin2Choice}_n$ instances is completely independent of each other, we let all the challenges be shared between them, provided that the random oracle which generates the challenges takes into account all the filled parts of the common transcript.

The final $l \times \mathtt{zkVC}_n$ calls on the 'invocation stack' (19) are needed only to prove that each of l vector commitments, namely each element of

$$\{Z_k + \delta_1 r_k F_k\}_{k=0}^{l-1}$$

is constructed over the common set of orthogonal generators

$$\{P_i + \delta_1 c_i Q_i\}_{i=0}^{n-1}.$$

Therefore, we can replace these $l \times \mathtt{zkVC}_n$ calls with one call to $\mathtt{zkMVC}_{l,n}$, thus making the 'invocation stack' (19) look as

$$l \times \mathtt{zkLin2Choice}_n \hookrightarrow l \times \mathtt{zkSVC}_{2,n} \hookrightarrow\ \mathtt{zkMVC}_{l,n}\,.$$

1.2.9 SIGNATURE EFLRSL

The EFLRS1 signature scheme in Figure 10 we constructed in Section 1.2.6 is, in sum, about that prover builds a key image I of type (15), then publishes it, then verifier sends a challenge ζ. Then using the one-out-of-many proof of membership $\texttt{zkLin2Choice}_n$ the prover convinces the verifier that Z built by the formula (16) belongs to the decoy set built by the formula (14), namely, the set of pairs

$$(\,\mathbf{P} + \zeta\mathbf{U},\, \mathbf{Q}\,)\,,\ \text{where}\ \mathbf{U} = \{\mathcal{H}_{\mathbf{point}}(P_i)\}_{i=0}^{n-1}$$

Suppose prover publishes a vector of l key images

$$\mathbf{I} = \{I_k\}_{k=0}^{l-1}\,,$$

of type (15) each, corresponding to l different indices $\mathbf{s} = \{s_k\}_{k=0}^{l-1}$ which we call actual signing indices or actual signers in the ring. The corresponding signing private keys $\mathbf{x} = \{x_k\}_{k=0}^{l-1}$ are, of course, assumed to be known to the prover. Taking random ζ both prover and verifier construct l values of Z by the formula (16), i.e. they construct the vector

$$\mathbf{Z} = \{Z_k\}_{k=0}^{l-1} = \{G\}^l + \zeta\,\mathbf{I} = \{G + \zeta I_k\}_{k=0}^{l-1}\,,$$

and also they build the same decoy set by the formula (14). After that, as the last step, they play the $\texttt{zkLin2Choice}_n$ one-out-of-many proof protocol l times for the decoy set and for each $Z_k,\ k \in [0\,\ldots\,l-1]$, we depict this as

$$l \times \texttt{zkLin2Choice}_n\,.$$

Although, instead of playing the one-out-of-many proof protocol l times, they might as well play the many-out-of-many proof protocol $\texttt{zkLin2mChoice}_{n,l}$ once. By doing so, they obtain a threshold version of the signature, which we call EFLRSL (Efficient linkable ring signature for l actual signers), its scheme

$$\texttt{EFLRSL.SignAndVerify}_{l,n}(\mathsf{M}, \mathbf{P}; \mathbf{s}, \mathbf{x})$$

is shown in Figure 14. Its size is $2\log_2(n) + 3l + 3$. The key image vector $\{I_k\}_{k=0}^{l-1}$ is taken into account in the calculation. Ring $\mathbf{P}$ is as usual assumed to be known beforehand for both prover and verifier.

1.2.10 HIDDEN AMOUNT EXTENSION

We have created the EFLRS1 signature using the $\texttt{zkLin2Choice}_n$ proof of membership protocol, which selects one element from a set of elements, or in other words, which proves the relation (12). Also, by runnining multiple parallel instances of $\texttt{zkLin2Choice}_n$ and optimizing their execution, we have created EFLRSL, which is a threshold version of the signature EFLRS1.

Suppose that the EFLRSL signature is used in a blockchain, where besides the public key P each address contains an additional element A storing some encrypted value called hidden amount. Formally, let's assume that each address is a pair (P, A) such that

$$A = bB + dD,$$

where B and D are independent fixed orthogonal generators, b is an amount, and d is this amount's blinding factor. That is, A hides the amount b protecting it from revealing with white noise d.

Now we want to enhance the EFLRSL signature, so that it will also be a zero-knowledge argument of that all b's behind A's of actual signers sum up to a given hidden amount $A^{\mathbf{sum}}$. We will describe the main idea of how we are going to do this, however, first, let's define denotations.

- Let a ring be composed of n pairs

$$\{(P_i, A_i)\}_{i=0}^{n-1}\,,\ \text{where}\ \mathbf{P} = \{P_i\}_{i=0}^{n-1}\ \text{and}\ \mathbf{A} = \{A_i\}_{i=0}^{n-1}\,. \tag{20}$$

  In the honest case we assume the following two assertions hold, with the scalars known to at least one of the players, for each $i \in [0\,\ldots\,n-1]$

$$P_i = p_i G\,, \tag{21}$$

$$A_i = b_i B + d_i D\,. \tag{22}$$

  In general, as usual, we assume a dishonest case, i.e. that the equalities (22) and (21) may not hold and, moreover, that some or all P_i's and A_i's in the ring may be adversarially chosen. However, from now on we will assume that some valid proofs of (22) for all A_i's in the ring have already been provided and verified. That is, we will assume that the relation (22) is satisfied for all A_i's participating in the ring. With this precondition, in the worst case, P_i's can have adversarially chosen p_i's or can have unknown relation to G, whereas A_i's can only have adversarially chosen b_i's and d_i's.

- $\mathcal{P}$ has two vectors, $\mathbf{s} = \{s_k\}_{k=0}^{l-1}$ and $\mathbf{x} = \{x_k\}_{k=0}^{l-1}$, which contain actual signing indices and corresponding private keys such that
$$P_{s_k} = x_k G.$$
- $\mathcal{P}$ and $\mathcal{V}$ have an element $A^{\mathbf{sum}}$ which represents total hidden amount, $\mathcal{P}$ knows its opening
$$A^{\mathbf{sum}} = b^{\mathbf{sum}} B + d^{\mathbf{sum}} D. \tag{23}$$
- $\mathcal{P}$ signs with $\mathbf{x}$, in doing so it knows the actual signer hidden amount openings
$$\mathbf{A}^{\mathbf{in}} = \{A_{s_k}\}_{k=0}^{l-1} = \{b_{s_k} B + d_{s_k} D\}_{k=0}^{l-1} \subseteq \mathbf{A}, \text{ where } \mathcal{P} \text{ knows all } b_{s_k}\text{'s and } d_{s_k}\text{'s.}$$
- Along with the signature the prover must provide a proof of the next balance
$$b^{\mathbf{sum}} = \sum_{k=0}^{l-1} b_{s_k}\,, \tag{24}$$
namely, a proof that the sum of hidden amounts of the signing addresses is equal to $A^{\mathbf{sum}}$ with the accuracy of a blinding component proportional to D.

Our idea of integrating the hidden amounts into the signature is that prover will send to verifier a vector of 'temporary' elements $\mathbf{A}^{\mathbf{tmp}} = \{A_k^{\mathbf{tmp}}\}_{k=0}^{l-1}$ and prove the following three additional assertions

1. For each $k \in [0 \dots l-1]$ the 'temporary' element $A_k^{\mathbf{tmp}}$ is equal to s_k-th hidden amount A_{s_k} in the ring to the accuracy of blinding component proportional to H, where H is a blinding generator of the signature. That is,
$$A_k^{\mathbf{tmp}} = A_{s_k} + f_{k\,H} H\,. \tag{25}$$
Here prover is free to randomly sample all the factors $f_{k\,H}$.
2. All $A_k^{\mathbf{tmp}} \in \mathbf{A}^{\mathbf{tmp}}$ sum up to $A^{\mathbf{sum}}$ to the accuracy of a linear by H and D component. That is,
$$A^{\mathbf{sum}} = \sum_{k=0}^{l-1} A_k^{\mathbf{tmp}} + f_H H + f_D D\,. \tag{26}$$
Here prover is free to randomly sample the factor f_D, and at the same time it is able to pick f_H as
$$f_H = -\sum_{k=0}^{l-1} f_{k\,H}\,. \tag{27}$$
3. If $A^{\mathbf{sum}}$ decomposes into a weighted sum of the generators B, H, and D with known weights, then the weight of the generator H in the decomposition is zero, i.e. the form (23) is fulfilled in such a case.

Since the signature should not reveal the signing s_k's and an since an observer should not be able to determine which A_{s_k}'s were summed up, we introduce $A_k^{\mathbf{tmp}}$'s as the explicit replacements of the corresponding A_{s_k}'s. With $A_k^{\mathbf{tmp}}$'s the observer still cannot determine anything due to the fact that each $A_k^{\mathbf{tmp}}$ has a blinding componenet proportional to H, namely, $f_{k_H} H$, where f_{k_H} is randomly sampled by prover.

From the assertions 1, 2, and from the equalities (22), (25), (26) it follows that verifier is convinced that there are the following decompositions of $A^{\mathbf{sum}}$ with known weights

$$\begin{aligned} A^{\mathbf{sum}} = \sum_{k=0}^{l-1} A_k^{\mathbf{tmp}} + f_H H + f_D D = \\ \sum_{k=0}^{l-1} (A_{s_k} + f_{k\,H} H) + f_H H + f_D D = \\ \sum_{k=0}^{l-1} (b_{s_k} B + d_{s_k} D + f_{k\,H} H) + f_H H + f_D D = \\ \sum_{k=0}^{l-1} b_{s_k} B + \left(\sum_{k=0}^{l-1} f_{k\,H} + f_H\right) H + \left(\sum_{k=0}^{l-1} d_{s_k} + f_D\right) D\,, \end{aligned} \tag{28}$$

where the scalars f_H and f_D are chosen by prover. If prover chooses the scalar f_H according to the equality (27), then the H's component of $A^{\mathbf{sum}}$ is equal to zero, i.e.

$$\left(\sum_{k=0}^{l-1} f_k H + f_H\right) H = 0\,.$$

Because of the assertion 3 the verifier is convinced that this is the case. Thus, the decomposition (28) for $A^{\mathbf{sum}}$ gets simplified to the decomposition

$$A^{\mathbf{sum}} = \sum_{k=0}^{l-1} b_{s_k} B + \left(\sum_{k=0}^{l-1} d_{s_k} + f_D\right) D\,,$$

which, taking into account the decomposition (23), proves the required equality (24).

Returning to the blockchain, having published the sets of output addresses and output hidden amounts in a transaction, prover signs it and simultaneously proves the equality (24), taking the sum of the output hidden amounts as $A^{\mathbf{sum}}$. Also, for each of the output hidden amounts the prover will have to give a range proof, however range proofs are beyond the scope of this paper; they can be implemented with known methods, for instance, with those described in [2, 3].

As for our assumption about the decompositions (22) for $\mathbf{A}$, it can be fulfilled by including in each transaction a proof that all the newly created output hidden amounts have these decompositions known to signer. Such a proof can be obtained in many ways, the good thing is that it is already included in the case if the range proofs as in [2, 3] are used.

1.2.11 SIMPLIFIED LIN2-2CHOICE LEMMA

To implement the idea outlined in Section 1.2.10 we need to somehow insert the hidden amounts $\mathbf{A}$, total hidden amount $A^{\mathbf{sum}}$, temporary elements $\mathbf{A}^{\mathbf{tmp}}$, and proofs of the assertions 1, 2, 3 from Section 1.2.10 into the signature scheme. Apparently, $\mathbf{A}$ can be added to the decoy set with random weighting, i.e. instead of the form (14) the decoy set entries might look something like (actually it will look a bit different)

$$(\, P_i + \zeta \mathcal{H}_{\mathbf{point}}(P_i) + \omega A_i\,,\, \ldots\,)\,,$$

where ω is an additional random weight. Also, by calling

$$\mathtt{zk2ElemComm}(D, H, A^{\mathbf{sum}} - \sum_{k=0}^{l-1} A_k^{\mathbf{tmp}}\;;\; f_D, f_H)$$

prover can convince verifier that $A^{\mathbf{sum}}$ equals to $\sum_{k=0}^{l-1} A_k^{\mathbf{tmp}}$ to the accuracy of a linear by H and D component, thus proving the assertion 2 from Section 1.2.10.

The assertion 3 from Section 1.2.10, in turn, can be obtained using an ideal hash to curve (in fact, to group) function. We define the generator H as a hash to curve of all the common inputs and transcript data written to the moment of H's first usage. This way the elements $\mathbf{A}$, $A^{\mathbf{sum}}$, B, D are included into H's pre-image. Thus, $A^{\mathbf{sum}}$ is prohibited from containing H in its decomposition.

The only remaining problem is how to convince verifier in the assertion 1 from Section 1.2.10, i.e. how to equate each $A_k^{\mathbf{tmp}}$ to the corresponding A_{s_k} to the accuracy of H. To solve this problem, we enchance the Lin2-Choice lemma protocol and prove the properties of the enchanced protocol in a new lemma called Lin2-2Choice.

To facilitate understanding, for the first we formulate a simplified version of the Lin2-2Choice lemma with its simplified protocol. This version proves, as usual, to the accuracy of H, that commitment Z is a weighted sum with prover knowing the weights of P_s and V_t, where P_s is a ring element under secret index s, and V_t is another ring element under publicly seen index t. Compared to the Lin2-Choice lemma, the simplified version of the Lin2-2Choice lemma protocol allows us to select a weighted sum of exactly two ring elements at once, not just one. We will see later what can be obtained from this.

So, the simplified version of the Lin2-2Choice lemma provides the argument

$$\mathtt{zkLin22sChoice}_{n,m}(\mathbf{P}, \mathbf{Q}, \mathbf{V}, \mathbf{W}, H, Z, t; s, p, v, \alpha)$$

shown in Figure 16 for the following relation

$$\mathcal{R} = \left\{ \begin{array}{l} \mathbf{P}, \mathbf{Q} \in \mathbb{G}^{n*}, \mathbf{V}, \mathbf{W} \in \mathbb{G}^{m*}, H \in \mathbb{G}^*, Z \in \mathbb{G}, t \in [0 \ldots m-1]; \\ s \in [0 \ldots n-1], p, v, \alpha \in \mathbb{F}_{\mathfrak{p}} \end{array} \;\middle|\; Z = pP_s + vV_t + \alpha H \right\}, \qquad (29)$$

where the vectors $\mathbf{P}, \mathbf{Q} \in \mathbb{G}^{n*}$, $\mathbf{V}, \mathbf{W} \in \mathbb{G}^{m*}$ are common prover and verifier input. All $2(n+m)$ elements in these four vectors are orthogonal to each other. The vectors $\mathbf{Q}$ and $\mathbf{W}$ are for technical purposes, while the vectors $\mathbf{P}$ and $\mathbf{V}$ are used to compose the element $Z = pP_s + vV_t$, where s, p, v are secret, and t is public. The protocol $\texttt{zkLin22sChoice}_{n,m}$ is constructed as follows.

- $\mathcal{P}$ hands over the following pair of elements to $\mathcal{V}$

$$F \text{ and } E\,. \tag{30}$$

- $\mathcal{V}$ generates a set of $n+m$ challenges $\{c_i\}_{i=0}^{n+m-1}$.
- $\mathcal{P}$ and $\mathcal{V}$ construct a decoy set of two parts and of size $n+m$. The first part of the decoy set, of size n, contains the following triplets

$$\{(P_i,\, c_i Q_i,\, 0)\}_{i=0}^{n-1}\,, \tag{31}$$

  whereas the second part, of size m, contains the following ones

$$\{(V_i,\, 0,\, c_{n+i} W_i)\}_{i=0}^{m-1}\,. \tag{32}$$

- $\mathcal{P}$ replies with a scalar r, and then the following two elements are constructed

$$rF\,,\; c_{n+t}E\,. \tag{33}$$

- As the last step, $\mathcal{P}$ and $\mathcal{V}$ play $\texttt{zkSVC}_{3,(n+m)}$ and thus $\mathcal{V}$ gets convinced that $\mathcal{P}$ knows weights of the following decompositions

$$\begin{cases} Z = \text{lin}(\mathbf{P}, \mathbf{V}) \\ F = \text{lin}(\mathbf{Q}) \\ E = \text{lin}(\mathbf{W}) \end{cases}\,. \tag{34}$$

Here we omit mentioning blinding with H as an apparent procedure, which is always implied performed before transmitting elements from prover to verifier.

An informal explanation of the $\texttt{zkLin22sChoice}_{n,m}$ protocol is that considering the triplet of elements

$$(Z,\, rF,\, c_{n+t}E) \tag{35}$$

we prove with $\texttt{zkSVC}_{3,(n+m)}$ that the first, second, and third elements of the triplet (35) are linear combinations with the same coefficients of $n+m$ elements of, respectively, the first, second, and third dimensions of the decoy set composed of the parts (31) and (32). We observe that thereby all steps of the $\texttt{zkLin2Choice}_n$ and $\texttt{zkLin2Choice}_m$ protocols are actually performed for Z's 'projections' on $\mathbf{P}$ and on $\mathbf{V}$, respectively. That is, we observe that

$$Z = Z_P + Z_V,\text{ where } Z_P = \text{lin}(\mathbf{P}),\; Z_V = \text{lin}(\mathbf{V})\,. \tag{36}$$

Thus, we find out that all the steps of the Lin2-Choice lemma protocol have been performed for

- Z_P and the first part of the decoy set comprising n triples (31). The actual index s remains hidden because the response r is randomized, as in the Lin2-Choice lemma protocol.
- Z_V and the second part of the decoy set comprising m triples (32). The actual index t in this part is not hidden because the 'reply' c_{n+t} clearly reveals it. Nevertheless, this does not wreck the Lin2-Choice lemma argument, just makes it non-zero-knowledge by t.

Hence, by the Lin2-Choice lemma, verifier is convinced that the following holds for prover

$$\begin{cases} Z_P \sim P_s & \text{, where } s \text{ is secret} \\ Z_V \sim V_t & \text{, where } t \text{ is public} \end{cases}\,, \tag{37}$$

and therefore $Z = pP_s + vV_t$ for some p and v known to prover.

1.2.12 MULTIPLE SIMMETRIC VECTOR COMMITMENTS

Again, we need one more auxiliary zero-knowledge protocol.

$$\texttt{zkMSVC}_{l,3,n}(\mathbf{P}, \mathbf{Q}, \mathbf{R}, H, \mathbf{Z}, \mathbf{F}, \mathbf{E};\; \mathfrak{a}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})$$

shown in Figure 17 proves the same thing as l simultaneously played instances of the $\texttt{zkSVC}_{3,n}$ protocol prove. This is a protocol for the following relation

$$\mathcal{R} = \left\{ \begin{array}{l} \mathbf{P} \in \mathbb{G}^{n*}, \mathbf{Q}, \mathbf{R} \in \mathbb{G}^n, H \in \mathbb{G}^*, \mathbf{Z}, \mathbf{F}, \mathbf{E} \in \mathbb{G}^l; \\ \mathfrak{a} \in \mathbb{F}_{\mathfrak{p}}^{l \times n}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma} \in \mathbb{F}_{\mathfrak{p}}^{l} \end{array} \middle| \begin{array}{l} \mathbf{Z} = \mathfrak{a} \cdot \mathbf{P} + \boldsymbol{\alpha} \cdot H \ \wedge \\ \mathbf{F} = \mathfrak{a} \cdot \mathbf{Q} + \boldsymbol{\beta} \cdot H \ \wedge \\ \mathbf{E} = \mathfrak{a} \cdot \mathbf{R} + \boldsymbol{\gamma} \cdot H \end{array} \right\}, \tag{38}$$

where all generators $\mathbf{P}, \mathbf{Q}, \mathbf{R}, H$ are orthogonal to each other, and for which the other accompanying requirements are the same as for the relation (7) in Section 1.2.4.

We implement this protocol using random weighting, defining the following two vectors with random scalars δ_1 and δ_2

$$\begin{aligned} \mathbf{X} &= \mathbf{P} + \delta_1 \mathbf{Q} + \delta_2 \mathbf{R} \\ \mathbf{Y} &= \mathbf{Z} + \delta_1 \mathbf{F} + \delta_2 \mathbf{E}, \end{aligned}$$

and invoking the $\texttt{zkMVC}_{l,n}$ protocol for them. Thus, we get a proof for the relation (38) at the price (i.e., size) of one protocol $\texttt{zkMVC}_{l,n}$, and hence at the price of one $\texttt{zkVC}_n$.

1.2.13 LIN2-2CHOICE LEMMA

We can now construct the protocol

$$\texttt{zkLin22Choice}_{l,n,m}(\mathbf{P}, \mathbf{Q}, \mathbf{V}, \mathbf{W}, H, \mathbf{Z}; \mathbf{s}, \mathbf{p}, \mathbf{v}, \boldsymbol{\alpha})$$

shown in Figure 18, and prove the Lin2-2Choice lemma which states that $\texttt{zkLin22Choice}_{l,n,m}$ is a complete, zero-knowledge argument having computational witness-extended emulation for the relation

$$\mathcal{R} = \left\{ \begin{array}{l} \mathbf{P}, \mathbf{Q} \in \mathbb{G}^{n*}, \mathbf{V}, \mathbf{W} \in \mathbb{G}^{m*}, H \in \mathbb{G}^*, \mathbf{Z} \in \mathbb{G}^l; \\ \mathbf{s} \in [0 \ldots n-1]^l, \mathbf{p}, \mathbf{v}, \boldsymbol{\alpha} \in \mathbb{F}_{\mathfrak{p}}^{l} \end{array} \middle| \begin{array}{l} \forall k \in [0 \ldots l-1] : \\ Z_k = p_k P_{s_k} + v_k V_k + \alpha_k H \end{array} \right\}, \tag{39}$$

where the generators $\mathbf{P}, \mathbf{Q}, \mathbf{V}, \mathbf{W}, H$ are orthogonal to each other and $l \leqslant m$.

The relation (39) is essentially the relation (29) repeated for the first l elements of the decoy set's second part (32). Having such a correspondence between the relations (39) and (29), the $\texttt{zkLin22Choice}_{l,n,m}$ protocol is l instances of the protocol $\texttt{zkLin22sChoice}_{n,m}$ run in parallel, with an only refinement.

The refinement is that all the l instances of the $\texttt{zkLin22sChoice}_{n,m}$ protocol are played in sync and independently of each other (except for the common challenges, as for EFLRSL in Section 1.2.9) up to the last step, where l instances of $\texttt{zkSVC}_{3,n}$ are called. All these l calls of $\texttt{zkSVC}_{3,n}$, in turn, are replaced by one call to $\texttt{zkMSVC}_{l,3,n}$, which gives significant reduction in transcript size.

1.2.14 SIGNATURE EFLRSLWB (MULTRATUG) WITH HIDDEN AMOUNT SUM (BALANCE) PROOF

Given a ring of the form (20), i.e. $\{(P_i, A_i)\}_{i=0}^{n-1}$, prover provides l key images of the form (15) for different indices s in the ring. That is, knowing the secret indices $\mathbf{s} = \{s_k\}_{k=0}^{l-1}$ and corresponding private keys $\mathbf{x} = \{x_k\}_{k=0}^{l-1}$, prover publishes the key images

$$\mathbf{I} = \{I_k\}_{k=0}^{l-1} = \{x_k^{-1} \boldsymbol{\mathcal{H}}_{\mathbf{point}}(P_{s_k})\}_{k=0}^{l-1}. \tag{40}$$

Also, prover publishes an element $A^{\mathbf{sum}}$ and declares that, to the accuracy of a component proportional to the generator D, there holds

$$A^{\mathbf{sum}} = \sum_{k=0}^{l-1} A_{s_k} \,. \tag{41}$$

Next, prover and verifier play the following game. They choose an orthogonal blinding generator H as a hash to curve of everything they have in common, and the prover provides to the verifier a vector $\mathbf{A}^{\mathbf{tmp}}$ of l hidden amounts which correspond to the actual signing keys blinded with H, i.e.

$$\mathbf{A}^{\mathbf{tmp}} = \{A_{s_k} + \mu_k H\}_{k=0}^{l-1}, \ \text{where each } \mu_k \text{ is white noise.} \tag{42}$$

Then, prover sends to verifier a set of l what we call 'pseudo key images' $\mathbf{J}$, which are constructed as follows

$$\mathbf{J} = \{x_k^{-1} \boldsymbol{\mathcal{H}}_{\mathbf{point}}(H, A_k^{\mathbf{tmp}}) + \upsilon_k H\}_{k=0}^{l-1}, \ \text{where each } \upsilon_k \text{ is white noise.} \tag{43}$$

The term 'pseudo key image' comes from the fact that each J_k is structurally similar to I_k, except for that I_k takes $\mathcal{H}_{\mathbf{point}}$ of P_k, whereas J_k takes $\mathcal{H}_{\mathbf{point}}$ of $(H, \mathbf{A}^{\mathbf{tmp}}_k)$ and, additionally, is blinded. Apparently, J_k cannot be used as the real key image I_k for linking actual signers, since J_k is not unique due to the blinding. Note, that all the I_k's are published before H is generated, so they are independent of H even in the dishonest case.

In addition to this, prover and verifier generate one more orthogonal generator, K, as a hash to curve of everything they have in common to this moment. Now, using random weights ζ, ω, χ prover and verifier make vectors

$$\mathbf{X} = \mathbf{P} - \{K\}^n + \zeta \{\mathcal{H}_{\mathbf{point}}(P_i)\}_{i=0}^{n-1} - \omega\mathbf{A}\,, \tag{44}$$

$$\mathbf{V} = \{K\}^l + \omega\mathbf{A}^{\mathbf{tmp}} + \chi \{\mathcal{H}_{\mathbf{point}}(H, A^{\mathbf{tmp}}_k)\}_{k=0}^{l-1}\,, \tag{45}$$

$$\mathbf{Z} = \{G\}^l + \zeta\mathbf{I} + \chi\mathbf{J}\,, \tag{46}$$

where $\mathbf{I}, \mathbf{A}^{\mathbf{tmp}}, \mathbf{J}$ are built by prover, in the honest case, by formulas (40), (42), (43).

Then, prover and verifier call the $\mathtt{zkLin22Choice}_{l,n,m}$ protocol of the Lin2-2Choice lemma

$$\mathtt{zkLin22Choice}_{l,n,l}(\mathbf{X}, \mathbf{Q}, \mathbf{V}, \mathbf{W}, H, \mathbf{Z};\ \mathbf{s}, \mathbf{x}^{-1}, \mathbf{x}^{-1},\ \boldsymbol{\alpha}_H)\,, \tag{47}$$

where $\mathbf{Q}, \mathbf{W}$ are auxiliary orthogonal generators prepared in advance. Moreover, $\mathbf{Q}, \mathbf{W}$ are also orthogonal to $\mathbf{X}$ (44) and to $\mathbf{V}$ (45), this is accomplished by defining $\mathcal{H}_{\mathbf{point}}$ in such a way so that all its returned elements are orthogonal to $\mathbf{Q}, \mathbf{W}$. The vector $\boldsymbol{\alpha}_H$ comprises weights accumulated by the H components within the protocol.

Let's look inside the call (47)

- prover sends the vectors $\mathbf{F}, \mathbf{E} \in \mathbb{G}^{l*}$, they correspond to the elements F and E in the first step (30) of the protocol $\mathtt{zkLin22sChoice}_{n,m}$,
- verifier generates challenges $\mathbf{c} = \{c_i\}_{i=0}^{n+l-1}$,
- prover replies with $\mathbf{r} \in \mathbb{F}_{\mathfrak{p}}^{l*}$,
- both prover and verifier build vectors $\hat{\mathbf{F}} = \mathbf{r} \circ \mathbf{F}$ and $\hat{\mathbf{E}} = \mathbf{c}_{[n:(n+l)]} \circ \mathbf{E}$ with elements corresponding to the pair (33),
- then the decoy set of two parts of the forms (31) and (32) is made. The first part of the decoy set of size n unfolds as

$$\{(P_i - K + \zeta\mathcal{H}_{\mathbf{point}}(P_i) - \omega A_i,\ c_i Q_i,\ 0)\}_{i=0}^{n-1}\,, \tag{48}$$

  and the second part of size l unfolds as

$$\{(K + \omega A^{\mathbf{tmp}}_i + \chi\mathcal{H}_{\mathbf{point}}(H, A^{\mathbf{tmp}}_i),\ 0,\ c_{n+i} W_i)\}_{i=0}^{l-1}\,, \tag{49}$$

- both parts (48) and (49) comprising element triplets are placed in the three vectors $\hat{\mathbf{P}}, \hat{\mathbf{Q}}, \hat{\mathbf{R}} \in \mathbb{G}^{n+l}$, respectively. Then $\mathtt{zkMSVC}_{l,3,(n+l)}(\hat{\mathbf{P}}, \hat{\mathbf{Q}}, \hat{\mathbf{R}}, H, \mathbf{Z}, \hat{\mathbf{F}}, \hat{\mathbf{E}};\ \ldots)$ is called.

As a result, by the relation (39), for the vector $\mathbf{Z}$ defined in (46), for each $k \in [0 \ldots l-1]$, $Z_k \in \mathbf{Z}$ a proof (to the accuracy of H component) of the following chain of equalities is obtained

$$\begin{aligned}
Z_k = x_k^{-1} X_{s_k} + x_k^{-1} V_k \;=\; & \\
& x_k^{-1}(\, P_{s_k} - K + \zeta\mathcal{H}_{\mathbf{point}}(P_{s_k}) - \omega A_{s_k}\,) + x_k^{-1}(\, K + \omega A^{\mathbf{tmp}}_k + \chi\mathcal{H}_{\mathbf{point}}(H, A^{\mathbf{tmp}}_k)\,) \;= \\
& G + \zeta I_k + x_k^{-1}\omega(-A_{s_k} + A^{\mathbf{tmp}}_k) + \chi J_k \;= \\
& G + \zeta I_k + \chi J_k\,,
\end{aligned}$$

which, in its turn, proves the following three things. First, it proves that prover actually knows the signing private keys $\mathbf{x}$. Second, that the key images $\mathbf{I}$ are honestly built by the formula (40). These first two give us the signature just like EFLRSL. Third, that the equalities (25) hold for all the elements of $\mathbf{A}^{\mathbf{tmp}}$, otherwise there would be a summand with ω multiplier for Z_k.

Having the equalities (25) proven, recalling that $A^{\mathbf{sum}}$ cannot contain H in its decomposition by D, B, H, prover and verifier perform a Schnorr-like two-generator (H and D) commitment protocol for the difference $A^{\mathbf{sum}} - \sum_{k=0}^{l-1} A^{\mathbf{tmp}}_k$, namely, they call

$$\mathtt{zk2ElemComm}(D, H, A^{\mathbf{sum}} - \sum_{k=0}^{l-1} A^{\mathbf{tmp}}_k;\ f_D, f_H)\,,$$

thus obtaining a proof for the equality (41) to the accuracy of D. Recalling all the ring hidden amounts $\mathbf{A}$ already have the form (22) proven, by this they get a proof for the equality (24), i.e., the sought proof of the balance.

Thus, the log-size signature EFLRSLWB (Efficient linkable ring signature for l actual signers with balance proof) of size $2\log_2(n+l)+6l+6$, nicknamed Multratug, is created. Its scheme

$$\texttt{EFLRSLWB.SignAndVerify}_{l,n}(\mathsf{M}, \mathbf{P}, \mathbf{A}, A^{\mathbf{sum}}, D; \mathbf{s}, \mathbf{x}, d^{\mathbf{\Delta sum}})$$

is shown in Figure 21.

2 PRELIMINARIES

At the beginning of the formal presentation, we first outline the definitions, assumptions, and methods that we borrow from the base works. Also, we specify the notation we use in this paper. Then we provide the helper protocols that we will use in the following chapters. Concrete implementations of the helper protocols are not decisive; any other implementations can be used as long as they have the same properties. We show the concrete implementations only to calculate the size and complexity of the resulting signatures and to see if they can be optimized.

2.1 DEFINITIONS AND BASE WORKS

All our protocols in this paper, including the helpers schemes and signatures, perform for prime order groups without bilinear pairings in a trustless environment under the DDH assumption in the random oracle model. All the context, namely, the common reference string, trustless setup, assumptions, orthogonality definition, non-intaractivity through Fiat-Shamir heuristic, special honest verifier zero-knowledge (HVZK) and computational witness-extended emulation (WEE) proof methods, which we use, are exactly the same as in the work of Bünz et al. [2]. Using them as already well known, we do not quote or explain them in detail to save space, instead referring simply to the fact that they correspond to and can be taken from [2].

The same applies to the work of Chung et al. [3], which is based on [2]. The common reference string, setup, assumptions, orthogonality, non-intaractivity, HVZK and WEE proof methods are the same. Similarly, for our current work they can be taken from [3].

Our previous work [11] is also based on [2]. We conduct our research in [11] from the ground up, relying on mathematical logic and computational theory, as a consequence we define some terms and methods which are not used in [2] and [3]. Nevertheless, they are in agreement with [2] and [3], and, in sum, the common reference string, setup, assumptions, orthogonality, non-intaractivity, HVZK and WEE proof methods are the same too.

In this paper we stick entirely to the canvas of modern cryptography. We prove that the underlying proving system is HVZK and has WEE, as in [2, 3], and add a linking tag on top, which, although depriving the signature of the HVZK property, still leaves it simulable enough to apply proof methods for unforgeability, anonymity, and other useful properties from, for example, [5, 7].

For certainty, here we take the definitions from [2]. As a syntactic sugar we use the shorthands '$\sim$', 'lin', 'ort' defined in [11], although they can be resolved and omitted. Also, we use additive notation for exponentiation of group elements as in [11]. We record our protocols in a form inspired by [3]. We imply non-interactive Fiat-Shamir counterparts everywhere not mentioning them. In [11] we have collected existing definitions of the linkable ring signature, its variations and security models from various sources, and we use these definitions in this paper, with one slight difference in that what in [11] we call a generic linkable ring signature, here we simply call a linkable ring signature.

In general, in this paper we denote elements, scalars, vectors, indices, etc. in the usual way that most closely resembles the notation in our work [11]. To make reading easier, here is a list of basic notations

- $\bar{\mathsf{p}}$ denotes a big prime chosen to be the order of group $\mathbb{G}$ and of the corresponding scalar field $\mathbb{F}_{\bar{\mathsf{p}}}$.
- lowercase italic and lowercase Greek letters denote scalars in $\mathbb{F}_{\bar{\mathsf{p}}}$. Apostrophes, hats, and subscript indices could be appended, e.g. a, b_{12}, c', ζ', x_k. Also, lowercase italic letters can be used to designate integers used as indices or limits, e.g. n, i, j_1, s_k, this usage is clear from the context. Superscripts, e.g. ϵ^2, denote scalar exponentiation.
- a special case is a lowercase italic letter with a bold superscript, e.g. $d^{\mathbf{\Delta sum}}$, this denotes a regular scalar of $\mathbb{F}_{\bar{\mathsf{p}}}$, and the superscript in bold is purely explanatory.
- bold lowercase italic and bold lowercase Greek letters denote scalar vectors, e.g. $\mathbf{a}$, $\mathbf{b}$, $\boldsymbol{\alpha}$.
- bold lowercase Gothic letters denote scalar matrices, e.g. $\mathfrak{a}$.
- uppercase italic letters denote elements in $\mathbb{G}$. Apostrophes, hats, and subscript indices could be appended, e.g. A, B_{12}, D', P_{s_k}. Multiplication is used to denote element exponentiation by scalar, e.g. xG.

- a special case is an uppercase italic letter with a bold superscript, e.g. $A^{\mathbf{sum}}$, this denotes a regular element of $\mathbb{G}$, and the superscript in bold is purely explanatory.
- bold uppercase italic letters denote element vectors, e.g. $\mathbf{A}$, $\mathbf{P}$.
- $\bar{\mathsf{n}}$ denotes a maximum number of elements in a ring.
- The zero element in $\mathbb{G}$ and the zero scalar in $\mathbb{F}_{\bar{\mathsf{p}}}$ are denoted as 0; it is clear from context which set 0 belongs to. A vector of n zeros is denoted either as $\mathbf{0}^n$ or as $\{0\}^n$, both notations are equivalent.
- asterisk denotes that zero entries are excluded. That is, $\mathbb{F}^*_{\bar{\mathsf{p}}}$ means $\mathbb{F}_{\bar{\mathsf{p}}}$ without scalar 0, $\mathbb{G}^*$ means $\mathbb{G}$ without element 0. Substantially, for vectors, if $\mathbf{x} \in \mathbb{F}^{n*}_{\bar{\mathsf{p}}}$, $\mathbf{P} \in \mathbb{G}^{m*}$, then $\mathbf{x}$ and $\mathbf{P}$ are assumed containing no zeros in any position.
- star denotes Klein star. For instance, $\mathsf{M} \in \{0,1\}^\star$ means M is a bitstring.
- $\mathcal{H}_{\mathbf{scalar}}$ and $\mathcal{H}_{\mathbf{point}}$ are the ideal hash and hash to group element (to curve) functions respectively.
- the statement $\mathrm{ort}(S)$ means that all elements of the set S are orthogonal to each other. For example, if S is composed of images of $\mathcal{H}_{\mathbf{point}}$ on different pre-images, then $\mathrm{ort}(S)$.
- $A = \mathrm{lin}(\mathbf{B})$, where $\mathbf{B}$ is a non-empty vector of non-zero elements, means there is a known vector $\mathbf{x}$ such that $A = \langle \mathbf{x}, \mathbf{B} \rangle$. Syntactic sugar $A \sim B$ is equivalent to $A = \mathrm{lin}(\{B\})$.
- $\mathrm{nz}(\mathbf{B})$ means a subset of $\mathbf{B}$ containing all non-zero elements found in $\mathbf{B}$.
- access to vector and matrix items is performed using Python notation, as in [2]. Also, having a vector, say, $\mathbf{A}$, we imply that A_i means i-th item of $\mathbf{A}$, i.e. we imply that A_i is an alias of $\mathbf{A}_{[i]}$ and therefore $A_i = \mathbf{A}_{[i]}$. Often we write explicitly 'let $A_i \leftarrow \mathbf{A}_{[i]}$', although the equality is already implied.
- adding an element to a vector is denoted by comma, e.g. $\hat{\mathbf{X}} \leftarrow [\mathbf{X}, B]$ means that $\hat{\mathbf{X}} = [X_0, X_1, \ldots, X_{n-1}, B]$.
- writing our protocols we mix several assignment styles, they all are construed as imperative assignment. That is, for example, the expression 'let $x \leftarrow y$' means the same thing as 'assign $x = y$'. Typically we use 'let $x \leftarrow y$' to indicate that x gets the value of y and both won't change.
- as a rule, when we use the letter n to represent an integer, we assume that n is subject to an additional restriction, e.g., that n or $(n+1)$ must be a power of 2. Exact body of this restriction is entirely determined by concrete vector commitment argument in which this n is directly or indirectly used.

Using this notation, all the information available from the beginning to both $\mathcal{P}$ and $\mathcal{V}$ and known in all protocols by default is shown in Figure 1.

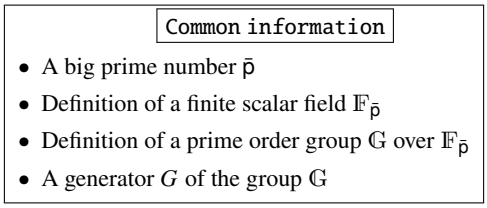

Figure 1: Information available to each party

2.2 TWO ELEMENT COMMITMENT

Theorem 1:
For two non-zero elements $X, H \in \mathbb{G}^*$ such that they are orthogonal to each other, for an element $Y \in \mathbb{G}$, the protocol `zk2ElemComm` *in Figure 2 is a complete, HVZK argument having WEE for the relation (2).*

Proof: Appendix A.
Overview: Section 1.2.1.

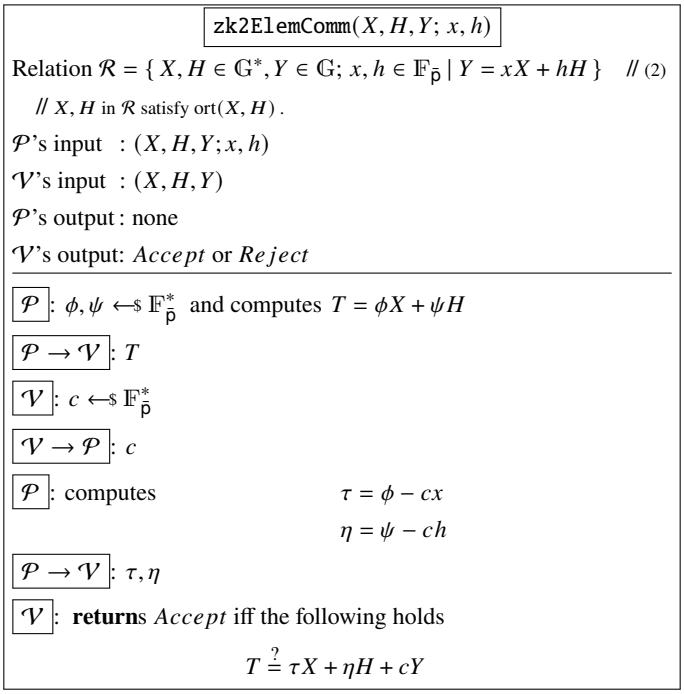

Figure 2: Zero-knowledge argument for two element commitment relation

2.3 BASIC VECTOR COMMITMENT

Theorem 2:
For $n \in \mathbb{N}^*$ such that n is a power of 2, for a vector of non-zero elements $\mathbf{X} \in \mathbb{G}^{n*}$, for a non-zero element $H \in \mathbb{G}^*$ such that all elements in $\mathbf{X} \cup \{H\}$ are orthogonal to each other, for an element $Y \in \mathbb{G}$, the protocol $\mathtt{zkVC}_n$ *in Figure 3 is a complete, HVZK argument having WEE for the relation (3).*

Proof: Appendix B.
Overview: Section 1.2.2.

$\boxed{\texttt{zkVC}_n(\mathbf{X}, H, Y;\ \mathbf{a}, \alpha)}$

Relation $\mathcal{R} = \{\ \mathbf{X} \in \mathbb{G}^{n*}, H \in \mathbb{G}^*, Y \in \mathbb{G};\ \mathbf{a} \in \mathbb{F}_{\bar{\mathsf{p}}}^n, \alpha \in \mathbb{F}_{\bar{\mathsf{p}}} \mid Y = \langle \mathbf{a}, \mathbf{X} \rangle + \alpha H\ \}$ // (3)

// $\mathbf{X}, H$ in $\mathcal{R}$ satisfy ort($\mathbf{X} \cup \{H\}$), n is a power of 2 everytime.

$\mathcal{P}$'s input : $(\mathbf{X}, H, Y; \mathbf{a}, \alpha)$

$\mathcal{V}$'s input : $(\mathbf{X}, H, Y)$

$\mathcal{P}$'s output : none

$\mathcal{V}$'s output: *Accept* or *Reject*

---

if $n > 1$ **then**

$\boxed{\mathcal{P}}$: $\beta, \gamma \leftarrow\!\$\ \mathbb{F}_{\bar{\mathsf{p}}}^*$ and computes $\hat{n} = n/2$

$$L = \left\langle \mathbf{a}_{[:\hat{n}]}, \mathbf{X}_{[\hat{n}:]} \right\rangle + \beta H$$
$$R = \left\langle \mathbf{a}_{[\hat{n}:]}, \mathbf{X}_{[:\hat{n}]} \right\rangle + \gamma H$$

$\boxed{\mathcal{P} \rightarrow \mathcal{V}}$: L, R

$\boxed{\mathcal{V}}$: $e \leftarrow\!\$\ \mathbb{F}_{\bar{\mathsf{p}}}^*$

$\boxed{\mathcal{V} \rightarrow \mathcal{P}}$: e

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: compute $\hat{\mathbf{X}} = e^{-1}\mathbf{X}_{[:\hat{n}]} + e\,\mathbf{X}_{[\hat{n}:]}$

$$\hat{Y} = Y + e^2 L + e^{-2} R$$

$\boxed{\mathcal{P}}$: computes $\hat{\mathbf{a}} = e\,\mathbf{a}_{[:\hat{n}]} + e^{-1}\mathbf{a}_{[\hat{n}:]}$

$$\hat{\alpha} = \alpha + e^2\beta + e^{-2}\gamma$$

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: run $\texttt{zkVC}_{\hat{n}}(\hat{\mathbf{X}}, H, \hat{Y};\ \hat{\mathbf{a}}, \hat{\alpha})$ // run recursively until n=1

else // n=1

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: let $X_0 \leftarrow \mathbf{X}_{[0]}$

and run $\texttt{zk2ElemComm}(X_0, H, Y;\ a_0, \alpha)$

endif

Figure 3: Zero-knowledge argument for vector commitment relation

2.4 RANDOM WEIGHTING FOR 3-TUPLES

Theorem 3:
For a non-zero element $P \in \mathbb{G}^*$, for a pair of elements $Q, R \in \mathbb{G}$, for a non-zero element $H \in \mathbb{G}^*$ such that all non-zero elements of the set $\{P, Q, R, H\}$ are orthogonal to each other and at least one of the two elements Q, R is non-zero, the protocol `zk3ElemRW` *in Figure 4 is a complete, HVZK argument having WEE for the relation (6).*

Proof: Appendix C.
Overview: 1.2.3.

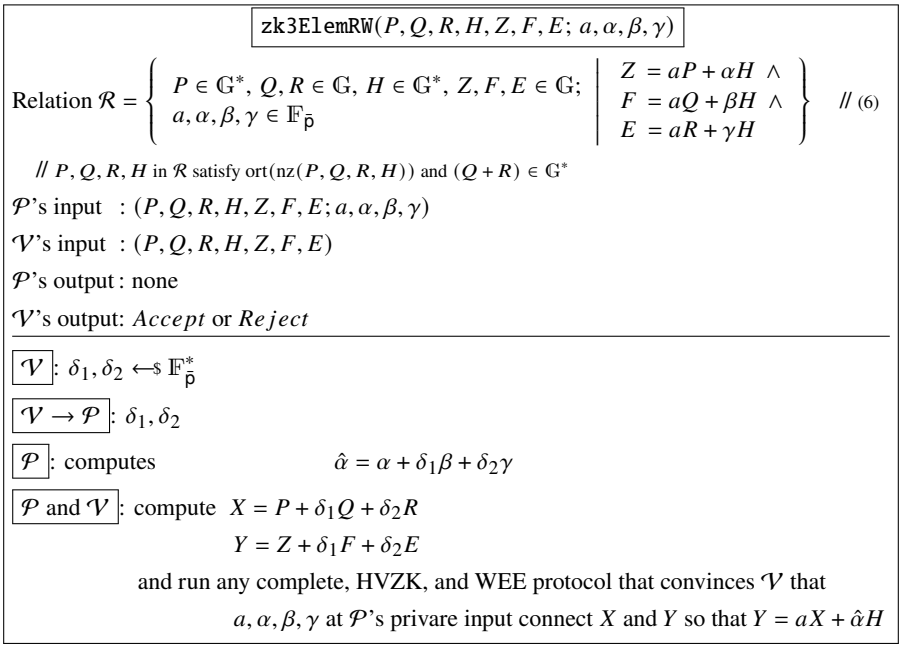

Figure 4: Zero-knowledge argument for two 3-tuples proportional to each other

2.5 SIMMETRIC VECTOR COMMITMENT

Theorem 4:
For $n \in \mathbb{N}^*$, for a vector of non-zero elements $\mathbf{P} \in \mathbb{G}^{n*}$, and for a pair of vectors of elements $\mathbf{Q}, \mathbf{R} \in \mathbb{G}^n$ such that $(\mathbf{Q} + \mathbf{R}) \in \mathbb{G}^{n*}$, for a non-zero element $H \in \mathbb{G}^*$ such that all non-zero elements in the set $\mathbf{P} \cup \mathbf{Q} \cup \mathbf{R} \cup \{H\}$ are orthogonal to each other, for three elements $Z, F, E \in \mathbb{G}$, the protocol $\mathtt{zkSVC}_{3,n}$ in Figure 5 is a complete, HVZK argument having WEE for the relation (7).

Proof: Appendix D.
Overview: 1.2.4.

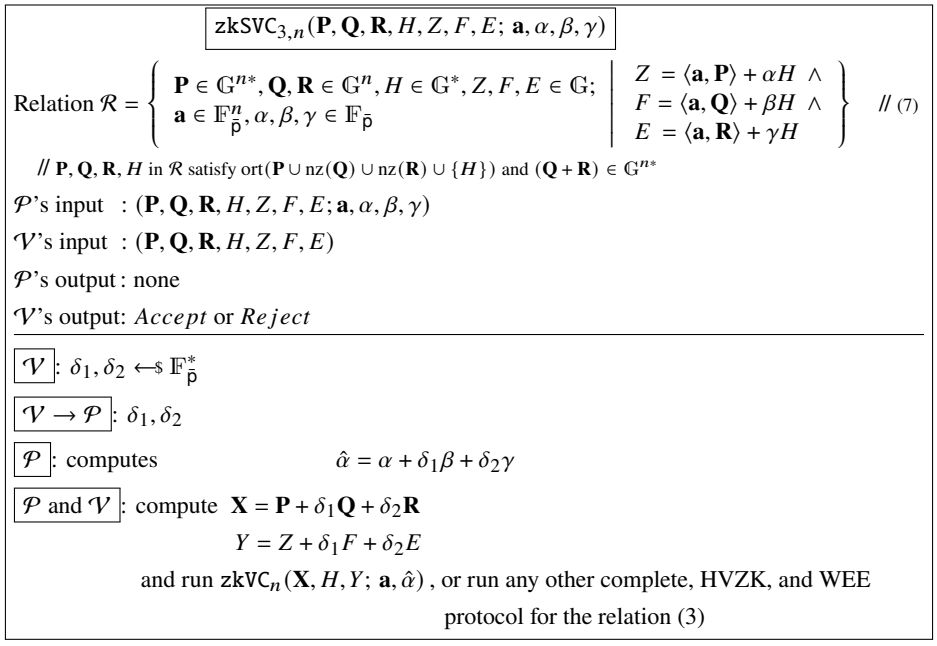

Figure 5: Zero-knowledge argument for 3 vector commitments with shared weights

As a special case of the $\texttt{zkSVC}_{3,n}$ protocol in Figure 5, we define the $\texttt{zkSVC}_{2,\mathrm{n}}$ protocol in Figure 6 for $\mathbf{R} = \mathbf{0}^n$, requiring for it that all elements of $\mathbf{Q}$ be non-zero.

$\boxed{\texttt{zkSVC}_{2,n}(\mathbf{P}, \mathbf{Q}, H, P, Q;\ \mathbf{a}, \alpha, \beta)}$

$\texttt{zkSVC}_{2,n}(\mathbf{P}, \mathbf{Q}, H, Z, F;\ \mathbf{a}, \alpha, \beta) = \texttt{zkSVC}_{3,n}(\mathbf{P}, \mathbf{Q}, \mathbf{0}^n, H, Z, F, 0;\ \mathbf{a}, \alpha, \beta, 0)$

// where $\mathbf{P}, \mathbf{Q} \in \mathbb{G}^{n*},\ H \in \mathbb{G}^*,\ Z, F \in \mathbb{G};\ \mathbf{a} \in \mathbb{F}_{\mathsf{p}}^n,\ \alpha, \beta, \gamma \in \mathbb{F}_{\mathsf{p}}$

Figure 6: Zero-knowledge argument for 2 vector commitments with shared weights

3 LINKABLE RING SIGNATURE

In this chapter we prove the Lin2-Choice lemma, which introduces 1-out-of-many proof of membership $\texttt{zkLin2Choice}_n$, and create a version of linkable ring signature for one actual signer, calling it EFLRS1.

3.1 LIN2-CHOICE LEMMA

Theorem 5 (Lin2-Choice lemma)**:**
For $n \in \mathbb{N}^*$, for two vectors of non-zero elements $\mathbf{P}, \mathbf{Q} \in \mathbb{G}^{n*}$, for a non-zero element $H \in \mathbb{G}^*$ such that all elements of the set $\mathbf{P} \cup \mathbf{Q} \cup \{H\}$ are orthogonal to each other, for an element $Z \in \mathbb{G}$, the protocol $\texttt{zkLin2Choice}_n$ in Figure 7 is a complete, HVZK argument having WEE for the relation (12).

Proof: Appendix E.
Overview: Section 1.2.5.

$\boxed{\texttt{zkLin2Choice}_n(\mathbf{P}, \mathbf{Q}, H, Z; s, p, \alpha)}$

Relation $\mathcal{R} = \left\{ \begin{array}{l} \mathbf{P}, \mathbf{Q} \in \mathbb{G}^{n*}, H \in \mathbb{G}^*, Z \in \mathbb{G}\,; \\ s \in [0 \ldots n-1], p, \alpha \in \mathbb{F}_{\mathsf{p}} \end{array} \middle|\ Z = pP_s + \alpha H \right\}$ // (12)

// $\mathbf{P}, \mathbf{Q}, H$ in $\mathcal{R}$ satisfy ort($\mathbf{P} \cup \mathbf{Q} \cup \{H\}$) .

$\mathcal{P}$'s input : $(\mathbf{P}, \mathbf{Q}, H, Z;\ s, p, \alpha)$

$\mathcal{V}$'s input : $(\mathbf{P}, \mathbf{Q}, H, Z)$

$\mathcal{P}$'s output : none

$\mathcal{V}$'s output: *Accept* or *Reject*

$\boxed{\mathcal{P}}$: $q, \beta \leftarrow\!\$\ \mathbb{F}_{\mathsf{p}}^*$ and assigns **if** $p = 0$ **then** $q = 0$ **endif**
$F = qQ_s + \beta H$

$\boxed{\mathcal{P} \to \mathcal{V}}$: F

$\boxed{\mathcal{V}}$: $\mathbf{c} \leftarrow\!\$\ \mathbb{F}_{\mathsf{p}}^{n*}$

$\boxed{\mathcal{V} \to \mathcal{P}}$: $\mathbf{c}$

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: compute $\hat{\mathbf{Q}} = \mathbf{c} \circ \mathbf{Q}$

$\boxed{\mathcal{P}}$: takes scalar c_s at index s in $\mathbf{c}$, that is, lets $c_s \leftarrow \mathbf{c}_{[s]}$,
samples $r \leftarrow\!\$\ \mathbb{F}_{\mathsf{p}}^*$,
assigns **if** $p \neq 0$ **then** $r = c_s p/q$ **endif**
$\hat{\beta} = r\beta$,
and lets $\mathbf{a} = \left\{ \begin{array}{l} a_s = p \quad \text{// that is, } p \text{ is at } s\text{'th position in one-hot } \mathbf{a} \text{ (or, if } p = 0, \text{ then } \mathbf{a} = \mathbf{0}^n) \\ a_i = 0 \text{ for all } i \in [0 \ldots n-1], i \neq s \end{array} \right.$

$\boxed{\mathcal{P} \to \mathcal{V}}$: r

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: let $\hat{F} \leftarrow rF$
and run $\texttt{zkSVC}_{2,n}(\mathbf{P}, \hat{\mathbf{Q}}, H, Z, \hat{F}; \mathbf{a}, \alpha, \hat{\beta})$

Figure 7: Zero-knowledge argument for one element choice relation

3.2 ADDITIONAL DEFINITIONS

To create the signature, we extend the common information in Figure 1 with the information in Figure 8. It is needed to ensure prover and verifier have identical definitions of hash $\mathcal{H}_{\mathbf{scalar}}$ and hash to group $\mathcal{H}_{\mathbf{point}}$ functions, as well as a common set of orthogonal generators $\mathbf{G}$.

The function $\mathcal{H}_{\mathbf{scalar}}$ models the random oracle. $\mathcal{H}_{\mathbf{point}}$ is used to generate a brand new element orthogonal to a set of existing elements. The predefined set $\mathbf{G}$ is used to reduce the signature verification complexity.

Additional common information

- Maximum number of elements in a ring $\bar{n}$
- Definition of an ideal hash finction $\mathcal{H}_{\mathbf{scalar}} : \{0,1\}^\star \to \mathbb{F}_{\bar{p}}^*$
- Definition of an ideal hash finction $\mathcal{H}_{\mathbf{point}} : \{0,1\}^\star \to \mathbb{G}^*$
- A vector of genearators $\mathbf{G} = \{G_0, G_1, G_2, \ldots, G_{\bar{n}-1}\} \in \mathbb{G}^{\bar{n}*}$
  such that for any set $\mathbf{H}$ of $\mathcal{H}_{\mathbf{point}}$ images on different pre-images there holds $\mathrm{ort}(\mathbf{H} \cup \{G\} \cup \mathbf{G})$

Figure 8: Additional information available to each party

All public keys of signatures can be known to all participants, and there are no additional restrictions on them. That is, in fact, we do not impose any rules on public keys, which is reflected in Figure 9.

Public keys

- There is an unlimited amount of public keys generated in the system.
- Each public key is considered visible to all parties, although this is not required.
- Each public key P is assumed to have the following relation to its private key x
  $$P = xG\,,$$
  although this is not required, i.e. there can exist public keys without any known x in this relation.
  There can exist adversarial public keys generated with purpose of breaking the signatures.

Figure 9: Public keys seen to all parties

3.3 SIGNATURE EFLRS1

Theorem 6:
For $n \in \mathbb{N}^*$, for a vector of non-zero elements $\mathbf{P} \in \mathbb{G}^{n*}$ which is considered as a ring of public keys, the protocol EFLRS1 in Figure 10 is a linkable ring signature with the following properties

1. *perfect correctness,*
2. *existential unforgeability against adaptive chosen message / public key attackers,*
3. *unforgeability w.r.t. insider corruption,*
4. *anonymity,*
5. *anonymity w.r.t. chosen public key attackers,*
6. *linkability,*
7. *non-frameability,*
8. *and non-frameability w.r.t. chosen public key attackers.*

Proof: Appendix F.
Overview: Section 1.2.6.

$\boxed{\texttt{EFLRS1.SignAndVerify}_{1,n}(\mathsf{M}, \mathbf{P}; s, x)}$

$\mathcal{P}$'s input : $(\mathsf{M} \in \{0,1\}^\star, \mathbf{P} \in \mathbb{G}^{n*};\ s \in [0 \ldots n-1], x \in \mathbb{F}^*_{\mathsf{p}})$

$\mathcal{V}$'s input : $(\mathsf{M} \in \{0,1\}^\star, \mathbf{P} \in \mathbb{G}^{n*})$

$\mathcal{P}$'s output : *Signature* // signature is a list of all $\mathcal{P} \to \mathcal{V}$ messages from this and nested protocols

$\mathcal{V}$'s output: *Accept* or *Reject*

$\boxed{\mathcal{P}}$: lets $P_s \leftarrow \mathbf{P}_{[s]}$,
assert $x \neq 0$
lets $p \leftarrow x^{-1}$
lets $I \leftarrow p\mathcal{H}_{\mathbf{point}}(P_s)$

$\boxed{\mathcal{P} \to \mathcal{V}}$: I

$\boxed{\mathcal{V}}$: $\epsilon, \zeta \leftarrow\$\ \mathbb{F}^*_{\mathsf{p}}$

$\boxed{\mathcal{V} \to \mathcal{P}}$: ϵ, ζ

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: **assert** all elements in $\mathbf{P}$ are non-zero and different
let $\mathbf{U} \leftarrow \{\mathcal{H}_{\mathbf{point}}(\mathbf{P}_{[i]})\}_{i=0}^{n-1}$,
$H \leftarrow \mathcal{H}_{\mathbf{point}}(\epsilon)$ // thus, ort$(H, \mathbf{G}, \mathbf{P}, \mathbf{U}, Z, I)$ holds
compute $\hat{\mathbf{P}} = \mathbf{P} + \zeta\mathbf{U}$
$Z = G + \zeta I$,
and run $\texttt{zkLin2Choice}_n(\hat{\mathbf{P}}, \mathbf{G}_{[:n]}, H, Z; s, p, 0)$

Figure 10: EFLRS1 signing and verification

In the signature schemes we always imply presence of one more procedure, `Link`, although we do not specify it explicitly. It is constructed trivially, as a comparison of key images I, just as in [7, 4, 11].

3.4 SIZE AND VERIFICATION COMPLEXITY

During execution of the $\texttt{EFLRS1.SignAndVerify}_{1,n}$ protocol a series of nested sub-protocols, up to the call of `zk2ElemComm`, is executed as shown in the top box in Figure 11. As a result, assuming that verifier postpones all calculations on its side until the end of the message exchange with prover, the verifier has only to check one expanded equality shown in Figure 11.

$\boxed{\texttt{SignAndVerify}_{1,n} \hookrightarrow \texttt{zkLin2Choice}_n \hookrightarrow \texttt{zkSVC}_{2,n} \hookrightarrow \texttt{zkVC}_n \hookrightarrow \texttt{zk2ElemComm}}$

// Function $\texttt{bitAtPos}(i, j)$ returns j-th bit of binary representation of i

$$c\left(G + \zeta I + \delta_1 rF + \sum_{j=0}^{\log_2(n)-1}(e_j^2 L_j + e_j^{-2} R_j)\right) + \eta H - T + \tau \sum_{i=0}^{n-1}\left(\prod_{j=0}^{\log_2(n)-1} e_j^{2\cdot\texttt{bitAtPos}(i,j)-1}\right)(P_i + \zeta U_i + \delta_1 c_i G_i) = 0$$

Figure 11: Unfolded equality for EFLRS1, verifier checks it

Table 1 shows the size and verification complexity of a batch of l EFLRS1 signatures that are created over a common ring of n public keys. We consider l signatures in order to compare the size and complexity against a threshold variant later. To get the size and verification complexity of single signature simply let $l = 1$.

To verify the batch, verifier combines l instances of the equality in Figure 11 using random weighting. As in [2, 3, 11], the verifier computes all the scalar weights which is considered negligibly time-consuming, and then performs single multi-exponentiation, resulting complexity is shown in Table 1.

Table 1: **EFLRS1** signature size and verification complexity

	Size	Verification complexity
EFLRS1	$l\left(2\log_2(n) + 6\right)$	$\boldsymbol{mexp}\left(3n + 2l\log_2(n) + 3l + 2\right) + (n+1)\mathbf{H}_{\mathbf{pt}}$

4 LINKABLE THRESHOLD RING SIGNATURE

To create a threshold variant of the signature we will define an auxiliary protocol $\texttt{zkMVC}_{l,n}$ that proves the same as l instances of $\texttt{zkVC}_n$ do. Then, by running l instances of $\texttt{zkLin2Choice}_n$ in parallel and substituting a $\texttt{zkMVC}_{l,n}$ call for l nested calls of $\texttt{zkVC}_n$ within them, we will get a many-out-of-many proof of membership, from which we will create the linkable threshold ring signature called EFLRSL.

4.1 MULTIPLE VECTOR COMMITMENTS

Theorem 7:
For $n, l \in \mathbb{N}^*$, for a vector of non-zero elements $\mathbf{X} \in \mathbb{G}^{n*}$, for a non-zero element $H \in \mathbb{G}^*$ such that all elements in $\mathbf{X} \cup \{H\}$ are orthogonal to each other, for a vector of elements $\mathbf{Y} \in \mathbb{G}^l$, the protocol $\texttt{zkMVC}_{l,n}$ in Figure 12 is a complete, HVZK argument having WEE for the relation (17).

Proof: Appendix G.
Overview: Section 1.2.7.

$\boxed{\texttt{zkMVC}_{l,n}(\mathbf{X}, H, \mathbf{Y};\ \mathfrak{a}, \boldsymbol{\alpha})}$

Relation $\mathcal{R} = \{\ \mathbf{X} \in \mathbb{G}^{n*}, H \in \mathbb{G}^*, \mathbf{Y} \in \mathbb{G}^l;\ \mathfrak{a} \in \mathbb{F}_{\mathfrak{p}}^{l \times n}, \boldsymbol{\alpha} \in \mathbb{F}_{\mathfrak{p}}^{l} \mid \mathbf{Y} = \mathfrak{a} \cdot \mathbf{X} + \boldsymbol{\alpha} \cdot H\ \}$ // (17)

// $\mathbf{X}, H$ in $\mathcal{R}$ satisfy ort$(\mathbf{X} \cup \{H\})$.

$\mathcal{P}$'s input : $(\mathbf{X}, H, \mathbf{Y}; \mathfrak{a}, \boldsymbol{\alpha})$

$\mathcal{V}$'s input : $(\mathbf{X}, H, \mathbf{Y})$

$\mathcal{P}$'s output : none

$\mathcal{V}$'s output: *Accept* or *Reject*

$\boxed{\mathcal{V}}$: $\boldsymbol{\xi} \leftarrow\!\$\ \mathbb{F}_{\mathfrak{p}}^{l*}$

$\boxed{\mathcal{V} \rightarrow \mathcal{P}}$: $\boldsymbol{\xi}$

$\boxed{\mathcal{P}}$: computes $\mathbf{a}^\intercal = \boldsymbol{\xi}^\intercal \cdot \mathfrak{a}$

$\alpha = \langle \boldsymbol{\xi}, \boldsymbol{\alpha} \rangle$

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: compute $Y = \langle \boldsymbol{\xi}, \mathbf{Y} \rangle$

and run $\texttt{zkVC}_n(\mathbf{X}, H, Y;\ \mathbf{a}, \alpha)$

Figure 12: Zero-knowledge argument for multiple vector commitments

4.2 MANY-OUT-OF-MANY PROOF

Theorem 8:
For $n \in \mathbb{N}^*$, for two vectors of non-zero elements $\mathbf{P}, \mathbf{Q} \in \mathbb{G}^{n*}$, for a non-zero element $H \in \mathbb{G}^*$ such that all elements of the set $\mathbf{P} \cup \mathbf{Q} \cup \{H\}$ are orthogonal to each other, for a vector of elements $\mathbf{Z} \in \mathbb{G}^l$, the protocol $\texttt{zkLin2mChoice}_{n,l}$ in Figure 13 is a complete, HVZK argument having WEE for the relation (18).

Proof: Appendix H.
Overview: Section 1.2.8.

$\boxed{\texttt{zkLin2mChoice}_{n,l}(\mathbf{P},\mathbf{Q},H,\mathbf{Z};\mathbf{s},\mathbf{p},\boldsymbol{\alpha})}$

Relation $\mathcal{R} = \left\{ \begin{array}{l|l} \mathbf{P},\mathbf{Q} \in \mathbb{G}^{n*}, H \in \mathbb{G}^{*}, \mathbf{Z} \in \mathbb{G}^{l}; & \forall k \in [0\ldots l-1]: \\ \mathbf{s} \in [0\ldots n-1]^{l}, \mathbf{p}, \boldsymbol{\alpha} \in \mathbb{F}_{\mathsf{p}}^{l} & Z_k = p_k P_{s_k} + \alpha_k H \end{array} \right\}$ // (18)

// $\mathbf{P},\mathbf{Q},H$ in $\mathcal{R}$ satisfy ort($\mathbf{P} \cup \mathbf{Q} \cup \{H\}$) .

$\mathcal{P}$'s input : $(\mathbf{P},\mathbf{Q},H,\mathbf{Z};\mathbf{s},\mathbf{p},\boldsymbol{\alpha})$

$\mathcal{V}$'s input : $(\mathbf{P},\mathbf{Q},H,\mathbf{Z})$

$\mathcal{P}$'s output : none

$\mathcal{V}$'s output: $Accept$ or $Reject$

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: allocate $\hat{\mathbf{X}} \in \mathbb{G}^{n}$, $\mathbf{Y} \in \mathbb{G}^{l}$, $\mathfrak{a} \in \mathbb{F}_{\mathsf{p}}^{l \times n}$, $\hat{\boldsymbol{\alpha}} \in \mathbb{F}_{\mathsf{p}}^{l}$,

and run the following block, depicted as foreach, in l parallel threads (with shared challenges),

using common $\hat{\mathbf{X}}, \mathbf{Y}, \mathfrak{a}, \hat{\boldsymbol{\alpha}}$

foreach $k \in [0\ldots l-1]$ // execute in parallel

let $(Z_k, s_k, p_k, \alpha_k) \leftarrow (\mathbf{Z}_{[k]}, \mathbf{s}_{[k]}, \mathbf{p}_{[k]}, \boldsymbol{\alpha}_{[k]})$,

run $\texttt{zkLin2Choice}_n(\mathbf{P},\mathbf{Q},H,Z_k;s_k,p_k,\alpha_k)$ without calling nested $\texttt{zkVC}_n(\mathbf{X},H,Y;\mathbf{a},\hat{\alpha})$ in it,

instead assign $\hat{\mathbf{X}} = \mathbf{X}$ // $\mathbf{X}$ is the same in all threads

$\mathbf{Y}_{[k]} = Y$

$\mathfrak{a}_{[k]} = \mathbf{a}$

$\hat{\boldsymbol{\alpha}}_{[k]} = \hat{\alpha}$.

endforeach

run $\texttt{zkMVC}_{l,n}(\hat{\mathbf{X}}, H, \mathbf{Y};\ \mathfrak{a}, \hat{\boldsymbol{\alpha}})$

Figure 13: Zero-knowledge argument for multiple element choice relation

4.3 SIGNATURE EFLRSL

Theorem 9:
For $n, l \in \mathbb{N}^*$ such that $l \leqslant n$, for a vector of non-zero elements $\mathbf{P} \in \mathbb{G}^{n*}$ which is considered as a ring of public keys, the protocol EFLRSL in Figure 14 is a linkable threshold ring signature with the following properties

1. *perfect correctness,*
2. *existential unforgeability against adaptive chosen message / public key attackers,*
3. *unforgeability w.r.t. insider corruption,*
4. *anonymity,*
5. *anonymity w.r.t. chosen public key attackers,*
6. *linkability,*
7. *non-frameability,*
8. *non-frameability w.r.t. chosen public key attackers.*

Proof: Appendix J.
Overview: Section 1.2.9.

$\boxed{\texttt{EFLRSL.SignAndVerify}_{l,n}(\mathsf{M}, \mathbf{P}; \mathbf{s}, \mathbf{x})}$

$\mathcal{P}$'s input : $(\mathsf{M} \in \{0,1\}^\star, \mathbf{P} \in \mathbb{G}^{n*}; \mathbf{s} \in [0 \ldots n-1]^l, \mathbf{x} \in \mathbb{F}_{\bar{\mathsf{p}}}^{l*})$

$\mathcal{V}$'s input : $(\mathsf{M} \in \{0,1\}^\star, \mathbf{P} \in \mathbb{G}^{n*})$

$\mathcal{P}$'s output : *Signature* // signature is a list of all $\mathcal{P} \to \mathcal{V}$ messages from this and nested protocols

$\mathcal{V}$'s output: *Accept* or *Reject*

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: **assert** all elements in $\mathbf{P}$ are non-zero and different

let $\mathbf{U} \leftarrow \{\mathcal{H}_{\mathbf{point}}(\mathbf{P}_{[i]})\}_{i=0}^{n-1}$

$\boxed{\mathcal{P}}$: allocates $\mathbf{I} \in \mathbb{G}^{l*}$, $\mathbf{p} \in \mathbb{F}_{\bar{\mathsf{p}}}^{l*}$,

initializes **foreach** $k \in [0 \ldots l-1]$

assert $\mathbf{x}_{[k]} \neq 0$

$\mathbf{p}_{[k]} = \mathbf{x}_{[k]}^{-1}$

lets $(s_k, p_k) \leftarrow (\mathbf{s}_{[k]}, \mathbf{p}_{[k]})$,

$\mathbf{I}_{[k]} = p_k \mathbf{U}_{[s_k]}$ // vector $\mathbf{I}$ is filled in here

endforeach

$\boxed{\mathcal{P} \to \mathcal{V}}$: $\mathbf{I}$

$\boxed{\mathcal{V}}$: **assert** all elements in $\mathbf{I}$ are different // here $\mathcal{V}$ makes sure there is no actual signer signing twice

$\epsilon, \zeta \leftarrow\$ \mathbb{F}_{\bar{\mathsf{p}}}^*$

$\boxed{\mathcal{V} \to \mathcal{P}}$: ϵ, ζ

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: let $H \leftarrow \mathcal{H}_{\mathbf{point}}(\epsilon)$ // thus, ort$(H, \mathbf{G}, \mathbf{P}, \mathbf{U}, \mathbf{Z}, \mathbf{I})$ holds

compute $\hat{\mathbf{P}} = \mathbf{P} + \zeta\mathbf{U}$,

$\mathbf{Z} = \{G\}^l + \zeta\mathbf{I}$

run $\texttt{zkLin2mChoice}_{n,l}(\hat{\mathbf{P}}, \mathbf{G}_{[:n]}, H, \mathbf{Z}; \mathbf{s}, \mathbf{p}, \{0\}^l)$

Figure 14: EFLRSL signing and verification

4.4 SIZE AND COMPLEXITY

An only equality that verifier has to check in order to verify authenticity of the EFLRSL signature is shown in Figure 15. The signature size and verification complexity are provided in Table 2.

$$\boxed{\texttt{SignAndVerify}_{l,n} \hookrightarrow \times\, \texttt{zkLin2Choice}_n \hookrightarrow l \times \texttt{zkSVC}_{2,n} \hookrightarrow \texttt{zkMVC}_{l,n} \hookrightarrow \texttt{zkVC}_n \hookrightarrow \texttt{zk2ElemComm}}$$

// Function $\texttt{bitAtPos}(i, j)$ returns j-th bit of binary representation of i

$$c\left(\sum_{k=0}^{l-1} \xi_k (G + \zeta I_k + \delta_1 r_k F_k) + \sum_{j=0}^{\log_2(n)-1} (e_j^2 L_j + e_j^{-2} R_j)\right) + \eta H - T + \\ + \tau \sum_{i=0}^{n-1} \left(\prod_{j=0}^{\log_2(n)-1} e_j^{2 \cdot \texttt{bitAtPos}(i,j)-1}\right)(P_i + \zeta U_i + \delta_1 c_i G_i) = 0$$

Figure 15: Unfolded equality for EFLRSL, verifier checks it

Table 2: **EFLRSL** signature size and verification complexity

	Size	Verification complexity
EFLRSL	$2\log_2(n) + 3l + 3$	$\boldsymbol{mexp}\left(3n + 2\log_2(n) + 2l + 3\right) + (n+1)\mathbf{H}_{\mathbf{pt}}$

Comparing Table 2 and Table 1, we find that the treshold variant of the signature is asymptotically l times more compact. Also, the verification of the treshold variant is asymptotically slightly faster.

5 LINKABLE THRESHOLD RING SIGNATURE WITH HIDDEN AMOUNT SUM PROOF

Now we are going to append a proof of the sum of hidden amounts to the EFLRSL signature described in Section 4.3. We assume that the signature ring has the form (20), and, additionally, that for all hidden amounts A_i in the ring there are some proofs of the decompositions (22) that are already verified. Both prover and verifier know the summary hidden amount $A^{\mathbf{sum}}$, and we want the prover to provide to the verifier a proof of the equalities (23), (24) along with the signature.

For this purpose, we need to extend the Lin2-Choice lemma (Theorem 5) protocol in Figure 7 with a part that will be responsible for the hidden amounts. We will introduce such an extension in Figure 16, and in the Simplified Lin2-2Choice lemma (Theorem 10) we will prove its properties as an one-out-of-many proof with an additional element. Next, like with the transition from $\texttt{zkLin2Choice}_n$ to $\texttt{zkLin2mChoice}_{n,l}$, we will proceed to the many-out-of-many proof in Figure 18.

In the Lin2-2Choice lemma (Theorem 12) we will prove properties of the protocol in Figure 18 as a many-out-of-many proof with additional elements. Based on this protocol we will construct the scheme EFLRSLWB aka Multratug as a linkable threshold ring signature combined with a proof of the sum of hidden amounts.

5.1 SIMPLIFIED LIN2-2CHOICE LEMMA

Theorem 10:
For $n, m \in \mathbb{N}^*$, for four vectors of non-zero elements $\mathbf{P}, \mathbf{Q} \in \mathbb{G}^{n*}$, $\mathbf{V}, \mathbf{W} \in \mathbb{G}^{m*}$, for a non-zero element $H \in \mathbb{G}^*$ such that all elements in $\mathbf{P} \cup \mathbf{Q} \cup \mathbf{V} \cup \mathbf{W} \cup \{H\}$ are orthogonal to each other, for an element $Z \in \mathbb{G}$, the protocol $\texttt{zkLin22sChoice}_{n,m}$ in Figure 16 is a complete, HVZK argument having WEE for the relation (29).

Proof: Appendix K.
Overview: Section 1.2.11.

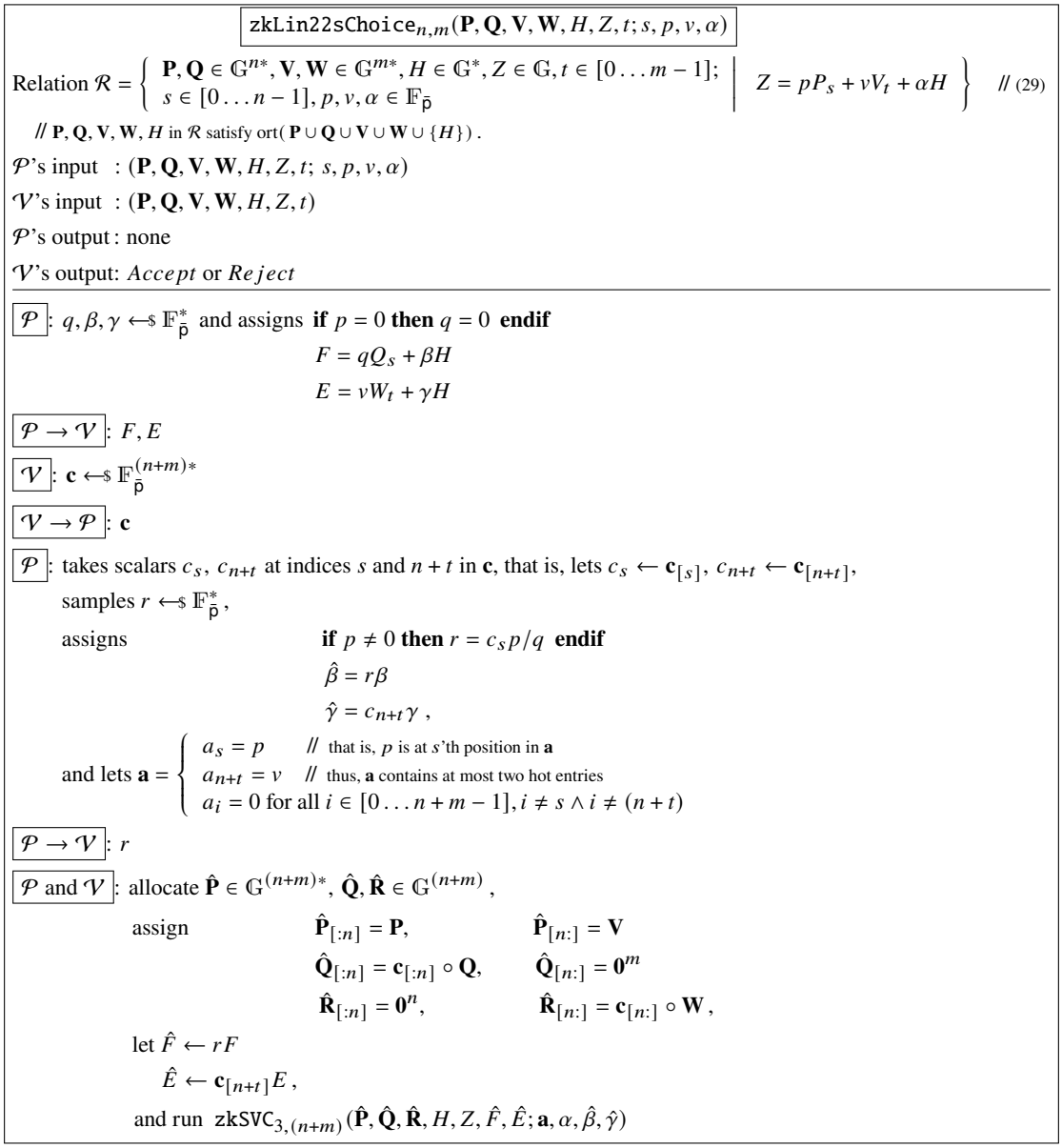

Figure 16: Simplified Lin2-2Choice lemma protocol, zero-knowledge argument for two-element choice relation

5.2 MULTIPLE SIMMETRIC VECTOR COMMITMENTS

To advance from the one-out-of-many proof to a many-out-of-many one, in Figure 17 we define a helper protocol.

Theorem 11:
For $n, l \in \mathbb{N}^*$, for a vector of non-zero elements $\mathbf{P} \in \mathbb{G}^{n*}$, and for a pair of vectors of elements $\mathbf{Q},\mathbf{R} \in \mathbb{G}^n$ such that $(\mathbf{Q}+\mathbf{R}) \in \mathbb{G}^{n*}$, for a non-zero element $H \in \mathbb{G}^*$ such that all non-zero elements in the set $\mathbf{P} \cup \mathbf{Q} \cup \mathbf{R} \cup \{H\}$ are orthogonal to each other, for three vectors of elements $\mathbf{Z},\mathbf{F},\mathbf{E} \in \mathbb{G}^l$, the protocol $\texttt{zkMSVC}_{l,3,n}$ in Figure 17 is a complete, HVZK argument having WEE for the relation (38).

Proof: Appendix L.
Overview: Section 1.2.12.

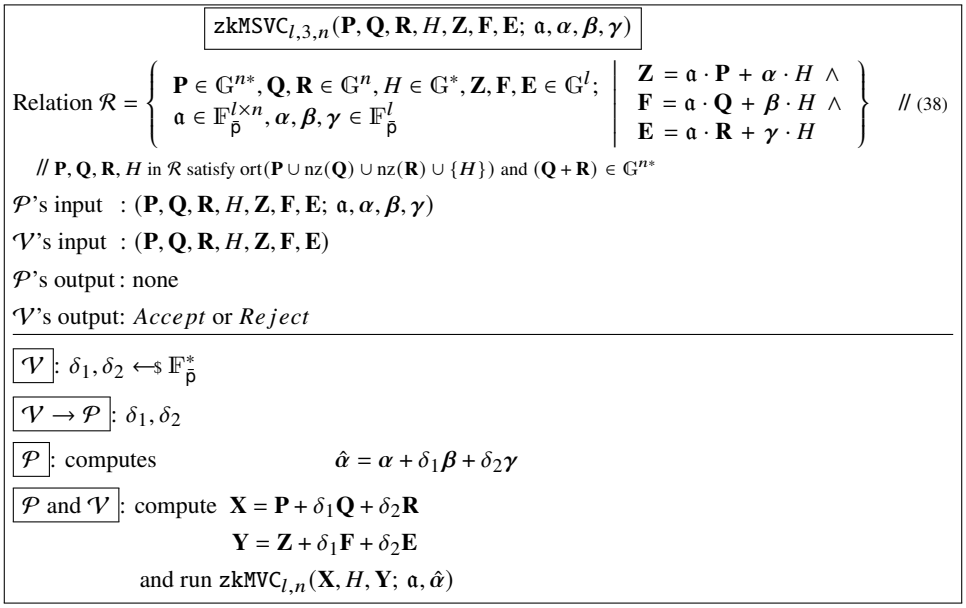

Figure 17: Zero-knowledge argument for multiple 3-vector commitments with shared weights

5.3 LIN2-2CHOICE LEMMA. MULTIPLE TWO-ELEMENT CHOICES

Theorem 12 (Lin2-2Choice lemma)**:**
For $n,m,l\in\mathbb{N}^*$ such that $l\leqslant m$, for four vectors of non-zero elements $\mathbf{P},\mathbf{Q}\in\mathbb{G}^{n*}$, $\mathbf{V},\mathbf{W}\in\mathbb{G}^{m*}$, for a non-zero element $H\in\mathbb{G}^*$ such that all elements in $\mathbf{P}\cup\mathbf{Q}\cup\mathbf{V}\cup\mathbf{W}\cup\{H\}$ are orthogonal to each other, for a vector of elements $\mathbf{Z}\in\mathbb{G}^l$, the protocol $\texttt{zkLin22Choice}_{l,n,m}$ in Figure 18 is a complete, HVZK argument having WEE for the relation (39)

Proof: Appendix M.
Overview: Section 1.2.13.

$\boxed{\texttt{zkLin22Choice}_{l,n,m}(\mathbf{P},\mathbf{Q},\mathbf{V},\mathbf{W},H,\mathbf{Z};\mathbf{s},\mathbf{p},\mathbf{v},\boldsymbol{\alpha})}$

Relation $\mathcal{R} = \left\{ \begin{array}{l} \mathbf{P},\mathbf{Q} \in \mathbb{G}^{n*}, \mathbf{V},\mathbf{W} \in \mathbb{G}^{m*}, H \in \mathbb{G}^{*}, \mathbf{Z} \in \mathbb{G}^{l}; \\ \mathbf{s} \in [0\ldots n-1]^{l}, \mathbf{p},\mathbf{v},\boldsymbol{\alpha} \in \mathbb{F}_{\bar{\mathsf{p}}}^{l} \end{array} \middle| \begin{array}{l} \forall k \in [0\ldots l-1]: \\ Z_k = p_k P_{s_k} + v_k V_k + \alpha_k H \end{array} \right\}$ // (39)

// $\mathbf{P},\mathbf{Q},\mathbf{V},\mathbf{W},H$ in $\mathcal{R}$ satisfy ort$(\mathbf{P} \cup \mathbf{Q} \cup \mathbf{V} \cup \mathbf{W} \cup \{H\})$.

$\mathcal{P}$'s input : $(\mathbf{P},\mathbf{Q},\mathbf{V},\mathbf{W},H,\mathbf{Z};\ \mathbf{s},\mathbf{p},\boldsymbol{\alpha})$

$\mathcal{V}$'s input : $(\mathbf{P},\mathbf{Q},\mathbf{V},\mathbf{W},H,\mathbf{Z})$

$\mathcal{P}$'s output : none

$\mathcal{V}$'s output: *Accept* or *Reject*

$\boxed{\mathcal{P}}$: $\mathbf{q},\boldsymbol{\beta},\boldsymbol{\gamma} \leftarrow\!\$\ \mathbb{F}_{\bar{\mathsf{p}}}^{l*}$, allocates $\mathbf{F},\mathbf{E} \in \mathbb{G}^{l*}$,

initializes **foreach** $k \in [0\ldots l-1]$

let $(s_k, p_k, v_k, \beta_k, \gamma_k) \leftarrow (\mathbf{s}_{[k]}, \mathbf{p}_{[k]}, \mathbf{v}_{[k]}, \boldsymbol{\beta}_{[k]}, \boldsymbol{\gamma}_{[k]})$,

if $p_k = 0$ **then** $\mathbf{q}_{[k]} = 0$ **endif** ,

let $q_k \leftarrow \mathbf{q}_{[k]}$,

$\mathbf{F}_{[k]} = q_k Q_{s_k} + \beta_k H$ // $\mathbf{F}$ is filled in, note random q_k's are nullified when $p_k = 0$

$\mathbf{E}_{[k]} = v_k W_k + \gamma_k H$ // $\mathbf{E}$ is filled in

endforeach

$\boxed{\mathcal{P} \rightarrow \mathcal{V}}$: $\mathbf{F},\mathbf{E}$

$\boxed{\mathcal{V}}$: $\mathbf{c} \leftarrow\!\$\ \mathbb{F}_{\bar{\mathsf{p}}}^{(n+m)*}$

$\boxed{\mathcal{V} \rightarrow \mathcal{P}}$: $\mathbf{c}$

$\boxed{\mathcal{P}}$: allocates $\hat{\boldsymbol{\beta}},\hat{\boldsymbol{\gamma}} \in \mathbb{F}_{\bar{\mathsf{p}}}^{l*}$, $\mathfrak{a} \in \mathbb{F}_{\bar{\mathsf{p}}}^{l\times(n+m)}$, samples $\mathbf{r} \leftarrow\!\$\ \mathbb{F}_{\bar{\mathsf{p}}}^{l*}$,

and initializes **foreach** $k \in [0\ldots l-1]$

let $c_{s_k} \leftarrow \mathbf{c}_{[s_k]}$

if $p_k \neq 0$ **then** $\mathbf{r}_{[k]} = c_{s_k} p_k / q_k$ **endif** // $\mathbf{r}$ is filled in here

endforeach ,

continues initialization $\hat{\boldsymbol{\beta}} = \mathbf{r} \circ \boldsymbol{\beta}$

$\hat{\boldsymbol{\gamma}} = \mathbf{c}_{[n:(n+l)]} \circ \boldsymbol{\gamma}$,

lets $\mathfrak{a} = \{a_{k\in[0\ldots l-1],i\in[0\ldots n+m-1]}\} = \left\{ \begin{array}{ll} a_{k,s_k} = p_k & \text{// that is, } p_k \text{ is at } s_k\text{'th position in } k\text{'th row} \\ a_{k,n+k} = v_k & \text{// that is, } v_k \text{ is at } (n+k)\text{'th position in } k\text{'th row} \\ a_{k,i} = 0 \ \text{ if } i \neq s_k \wedge i \neq (n+k) & \text{// zeros for all the rest} \end{array} \right.$

$\boxed{\mathcal{P} \rightarrow \mathcal{V}}$: $\mathbf{r}$

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: allocate $\hat{\mathbf{P}} \in \mathbb{G}^{(n+m)*}$, $\hat{\mathbf{Q}},\hat{\mathbf{R}} \in \mathbb{G}^{(n+m)}$,

assign $\hat{\mathbf{P}}_{[:n]} = \mathbf{P}$, $\hat{\mathbf{P}}_{[n:]} = \mathbf{V}$

$\hat{\mathbf{Q}}_{[:n]} = \mathbf{c}_{[:n]} \circ \mathbf{Q}$, $\hat{\mathbf{Q}}_{[n:]} = \mathbf{0}^m$

$\hat{\mathbf{R}}_{[:n]} = \mathbf{0}^n$, $\hat{\mathbf{R}}_{[n:]} = \mathbf{c}_{[n:]} \circ \mathbf{W}$,

let $\hat{\mathbf{F}} \leftarrow \mathbf{r} \circ \mathbf{F}$

$\hat{\mathbf{E}} \leftarrow \mathbf{c}_{[n:(n+l)]} \circ \mathbf{E}$,

and run $\texttt{zkMSVC}_{l,3,(n+m)}(\hat{\mathbf{P}},\hat{\mathbf{Q}},\hat{\mathbf{R}},H,\mathbf{Z},\hat{\mathbf{F}},\hat{\mathbf{E}};\ \mathfrak{a},\boldsymbol{\alpha},\hat{\boldsymbol{\beta}},\hat{\boldsymbol{\gamma}})$

Figure 18: Lin2-2Choice lemma protocol, zero-knowledge argument for multiple two-element choices relation

5.4 ADDITIONAL DEFINITIONS

Prior to constructing the signature with hidden amount sum proof, in Figure 19 we define how the hidden amounts are represented in the system.

Hidden amounts

- Each public key P is accompanied by a hidden amount A in the system. Each ring has the form (20).
- Each hidden amount A is assumed having the decomposition (22) by the predefined generators B, D, i.e.
$$A = bB + dD\,,$$
where b is the amount and d is the amount's blinding factor. It is assumed that for each hidden amount A in the sysyem there is a valid proof of the decomposition (22) for it.

Figure 19: Hidden amounts seen to all parties

We also need to supplement the common information available to all parties according to Figure 1 and Figure 8 with an extended set of predefined orthogonal generators, and to update the function $\mathcal{H}_{\mathbf{point}}$ one more time, as in Figure 20, so that it will respect orthogonality of the additional generators.

Updated common information

- A couple of genearators $B, D \in \mathbb{G}^*$ and the enlarged vector $\mathbf{G} = \{G_0, G_1, G_2, \ldots, G_{2\tilde{n}-1}\} \in \mathbb{G}^{2\tilde{n}*}$ such that for any set $\mathbf{H}$ of $\mathcal{H}_{\mathbf{point}}$ images on different pre-images there holds $\mathrm{ort}(\mathbf{H} \cup \{G, B, D\} \cup \mathbf{G})$
- $\mathcal{H}_{\mathbf{point}} : \{0,1\}^\star \to \mathbb{G}^*$ is updated so that the above $\mathrm{ort}(\mathbf{H} \cup \{G, B, D\} \cup \mathbf{G})$ holds.

Figure 20: Updated common information available to each party

5.5 SIGNATURE EFLRSLWB (MULTRATUG) WITH THE SUM PROOF

Theorem 13:
For $n, l \in \mathbb{N}^*$ such that $l \leqslant n$, for a vector of non-zero elements $\mathbf{P} \in \mathbb{G}^{n*}$ together with a vector of elements $\mathbf{A} \in \mathbb{G}^n$ which are considered a ring of (public key, hidden amount) pairs, for an element $A^{\mathbf{sum}}$, for a non-zero element D which is considered as a blinding generator for hidden amounts, the protocol in Figure 21 is a linkable threshold ring signature with the following properties

1. *perfect correctness,*
2. *existential unforgeability against adaptive chosen message / public key attackers,*
3. *unforgeability w.r.t. insider corruption,*
4. *anonymity,*
5. *anonymity w.r.t. chosen public key attackers,*
6. *linkability,*
7. *non-frameability,*
8. *non-frameability w.r.t. chosen public key attackers,*
9. *it is a proof of that $A^{\mathbf{sum}}$ is a sum of A's of the actual signing keys, to the accuracy of the blinding component proportional to D.*

Proof: Appendix O.
Overview: Section 1.2.14.

Note, Theorem 13 doesn't impose any requirement on elements of the vector $\mathbf{A}$ and on $A^{\mathbf{sum}}$, i.e., there is no assumption like (22) about their decompositions. At the same time, it's easy to see that if the property 9) is proven, then the balance (24) proof immediately follows from a proof of the decomposition (22) for all $A_k \in \mathbf{A}$. Therefore, if along with Multratug a proof of the decomposition (22) for all A_k's is obtained by any other means, then the balance (24) proof is thus obtained.

$\boxed{\texttt{EFLRSLWB.SignAndVerify}_{l,n}(\mathsf{M}, \mathbf{P}, \mathbf{A}, A^{\mathbf{sum}}, D; \mathbf{s}, \mathbf{x}, d^{\mathbf{\Delta sum}})}$

$\mathcal{P}$'s input : $(\mathsf{M} \in \{0,1\}^\star, \mathbf{P} \in \mathbb{G}^{n*}, \mathbf{A} \in \mathbb{G}^n, A^{\mathbf{sum}} \in \mathbb{G}, D \in \mathbb{G}^*; \mathbf{s} \in [0 \ldots n-1]^l, \mathbf{x} \in \mathbb{F}_{\bar{\mathsf{p}}}^{l*}, d^{\mathbf{\Delta sum}} \in \mathbb{F}_{\bar{\mathsf{p}}})$

$\mathcal{V}$'s input : $(\mathsf{M} \in \{0,1\}^\star, \mathbf{P} \in \mathbb{G}^{n*}, \mathbf{A} \in \mathbb{G}^n, A^{\mathbf{sum}} \in \mathbb{G}, D \in \mathbb{G}^*)$

$\mathcal{P}$'s output : *Signature* // signature is a list of all $\mathcal{P} \to \mathcal{V}$ messages from this and nested protocols

$\mathcal{V}$'s output: *Accept* or *Reject*

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: **assert** all elements in $\mathbf{P}$ are non-zero and different

let $\mathbf{U} \leftarrow \{\mathcal{H}_{\mathbf{point}}(\mathbf{P}_{[i]})\}_{i=0}^{n-1}$

$\boxed{\mathcal{P}}$: allocates $\mathbf{I} \in \mathbb{G}^{l*}$, $\mathbf{p} \in \mathbb{F}_{\bar{\mathsf{p}}}^{l*}$,

initializes **foreach** $k \in [0 \ldots l-1]$

assert $\mathbf{x}_{[k]} \neq 0$

$\mathbf{p}_{[k]} = \mathbf{x}_{[k]}^{-1}$

lets $(s_k, p_k) \leftarrow (\mathbf{s}_{[k]}, \mathbf{p}_{[k]})$,

$\mathbf{I}_{[k]} = p_k \mathbf{U}_{[s_k]}$ // vector $\mathbf{I}$ is filled in here

endforeach

$\boxed{\mathcal{P} \to \mathcal{V}}$: $\mathbf{I}$

$\boxed{\mathcal{V}}$: **assert** all elements in $\mathbf{I}$ are non-zero and different // $\mathcal{V}$ makes sure there is no zero I and no signer signing twice

$\epsilon \leftarrow\!\$\ \mathbb{F}_{\bar{\mathsf{p}}}^*$

$\boxed{\mathcal{V} \to \mathcal{P}}$: ϵ

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: let $H \leftarrow \mathcal{H}_{\mathbf{point}}(\epsilon)$ // thus, H is orthogonal to all known so far elements, i.e. $\mathrm{ort}(H, G, \mathbf{P}, \mathbf{A}, \mathbf{U}, \mathbf{I}, A^{\mathbf{sum}}, D)$

$\boxed{\mathcal{P}}$: $\boldsymbol{\mu}, \boldsymbol{\upsilon} \leftarrow\!\$\ \mathbb{F}_{\bar{\mathsf{p}}}^{l*}$, allocates $\mathbf{A}^{\mathbf{tmp}} \in \mathbb{G}^{l*}$, $\boldsymbol{\alpha} \in \mathbb{F}_{\bar{\mathsf{p}}}^{l*}$,

initializes **foreach** $k \in [0 \ldots l-1]$

lets $\mu_k \leftarrow \boldsymbol{\mu}_{[k]}$,

$\mathbf{A}^{\mathbf{tmp}}_{[k]} = \mathbf{A}_{[s_k]} + \mu_k H$ // $\mathbf{A}^{\mathbf{tmp}}$ is filled, amounts get double blinded (with D and with H)

$\boldsymbol{\alpha}_{[k]} = p_k \mu_k$ // $\boldsymbol{\alpha}$ is initialized here, it contains reduced $\mathbf{A}^{\mathbf{tmp}}$'s second blinding factors

endforeach

$\boxed{\mathcal{P} \to \mathcal{V}}$: $\mathbf{A}^{\mathbf{tmp}}$

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: let $\hat{\mathbf{U}} \leftarrow \{\mathcal{H}_{\mathbf{point}}(H, \mathbf{A}^{\mathbf{tmp}}_{[k]})\}_{k=0}^{l-1}$

$\boxed{\mathcal{P}}$: lets $\mathbf{J} \leftarrow \{p_k \hat{\mathbf{U}}_{[k]} + \upsilon_k H\}_{k=0}^{l-1}$ // vector $\mathbf{J}$ is initialized here, it contains 'pseudo key images' built using $\hat{\mathbf{U}}$

$\boxed{\mathcal{P} \to \mathcal{V}}$: $\mathbf{J}$

$\boxed{\mathcal{V}}$: **assert** all elements in $\mathbf{A}^{\mathbf{tmp}}, \mathbf{J}$ are non-zero and different // $\mathcal{V}$ makes sure $\hat{\mathbf{U}}$ is orthogonal and there is no zero J

$\hat{\epsilon}, \zeta, \omega, \chi \leftarrow\!\$\ \mathbb{F}_{\bar{\mathsf{p}}}^*$

$\boxed{\mathcal{V} \to \mathcal{P}}$: $\hat{\epsilon}, \zeta, \omega, \chi$

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: let $K \leftarrow \mathcal{H}_{\mathbf{point}}(\hat{\epsilon})$ // thus, $\mathrm{ort}(K, H, G, \mathbf{P}, \mathbf{A}, \mathbf{U}, \mathbf{I}, A^{\mathbf{sum}}, \mathbf{A}^{\mathbf{tmp}}, \hat{\mathbf{U}}, \mathbf{J})$ holds

allocate $\mathbf{X} \in \mathbb{G}^{n*}$, $\mathbf{V}, \mathbf{Z} \in \mathbb{G}^{l*}$, $S \in \mathbb{G}$,

assign $\mathbf{X} = \mathbf{P} - \{K\}^n + \zeta \mathbf{U} - \omega \mathbf{A}$, $\mathbf{V} = \{K\}^l + \omega \mathbf{A}^{\mathbf{tmp}} + \chi \hat{\mathbf{U}}$,

$\mathbf{Z} = \{G\}^l + \zeta \mathbf{I} + \chi \mathbf{J}$

assign $S = A^{\mathbf{sum}} - \sum_{k=0}^{l-1} \mathbf{A}^{\mathbf{tmp}}_{[k]}$

run $\texttt{zk2ElemComm}(D, H, S; d^{\mathbf{\Delta sum}}, -\sum_{k=0}^{l-1} \mu_k)$

run $\texttt{zkLin22Choice}_{l,n,l}(\mathbf{X}, \mathbf{G}_{[:n]}, \mathbf{V}, \mathbf{G}_{[n:(n+l)]}, H, \mathbf{Z}; \mathbf{s}, \mathbf{p}, \mathbf{p}, -\omega\boldsymbol{\alpha} + \chi\boldsymbol{\upsilon})$

Figure 21: Multratug signing and verification

5.6 SIZE AND COMPLEXITY

To verify the Multratug signature $\mathcal{V}$ needs only to check the equalities (*) and (**) in Figure 22. Combining the equalities (*) and (**) with random weighting and using multi-exponetiation technique $\mathcal{V}$ performs the verifiacation in the time shown in Table 3, where signature size is also shown.

$\texttt{SignAndVerify}_{l,n,u} \hookrightarrow \texttt{zkLin22Choice}_{l,n,l} \hookrightarrow \texttt{zkMSVC}_{l,3,(n+l)} \hookrightarrow \texttt{zkMVC}_{l,(n+l)} \hookrightarrow \texttt{zkVC}_{(n+l)} \hookrightarrow \texttt{zk2ElemComm}$

// Function $\texttt{bitAtPos}(i,\, j)$ returns j-th bit of binary representation of i

$$c\left(\sum_{k=0}^{l-1}\xi_k(G+\zeta I_k+\chi J_k+\delta_1 r_k F_k+\delta_2 c_{(n+k)}E_k)+\sum_{j=0}^{\log_2(n+l)-1}(e_j^2 L_j+e_j^{-2}R_j)\right)+\eta H-T+$$

$$+\tau\left(\sum_{i=0}^{n-1}\left(\prod_{j=0}^{\log_2(n+l)-1}e_j^{2\cdot\texttt{bitAtPos}(i,j)-1}\right)(P_i+\zeta U_i-\omega A_i+K+\delta_1 c_i G_i)+\right. \qquad (*)$$

$$\left.+\sum_{i=n}^{n+l-1}\left(\prod_{j=0}^{\log_2(n+l)-1}e_j^{2\cdot\texttt{bitAtPos}(i,j)-1}\right)(\omega A^{\mathbf{tmp}}_{(i-n)}+\chi\hat{U}_{(i-n)}-K+\delta_2 c_i G_i)\right)=0$$

and

$$\hat{\tau}D+\hat{\eta}H+\hat{c}S-\hat{T}=0 \qquad (**)$$

Figure 22: Multratug unfolded equality, verifier checks it

Table 3: **Multratug** signature size and verification complexity

	Size	Verification complexity
Multratug	$2\log_2(n+l)+6l+6$	$\boldsymbol{mexp}(\,4n+2\log_2(n+l)+7l+7\,)+(n+l+2)\mathbf{H_{pt}}$

6 IMPROVEMENTS

6.1 USING RING OF SIZE N·L

It is possible to slightly reduce the size of the Multratug scheme by not using the Lin2-2Choice lemma and instead repeating the ring l times, each time for its amount $A_k^{\mathbf{tmp}}$. In this case, after appropriate optimizations, the signature size would be

$$2\log_2(nl)+5l+O(1).$$

Nevertheless, we still prefer the version with the Lin2-2Choice lemma, because it is impossible to just repeat the ring l times, even using for each repetition its own independent generator, e.g. of the form $\mathcal{H}_{\mathbf{point}}(A_k^{\mathbf{tmp}})$. It would be necessary to add more generators to keep all the ring elements linearly independent of each other, which will correspondingly increase the verification time.

6.2 BATCH VERIFICATION

Verification of a batch of Multratug signatures can be accomplished with checking just one equality, by combining Figure 22's equalities (*) and (**) of all signatures in the batch using random weighting. In this case, the asymptotic verification complexity by ring size n under the multi-exponent decreases from $4n$ to $3n$ due to the fact, that all instances of the Multratug signature use the same vector of predefined generators $\mathbf{G}$.

The same can be stated about EFLRSL by referring to Figure 15 and finding there a decrease from $3n$ to $2n$ under the multi-exponent.

6.3 SAVING ONE ITEM IN ALL LOG-SIZE SCHEMES

It is possible to reduce by 1 the sizes of Multratug, EFLRSL, and other schemes that use the protocol $\texttt{zkVC}_n$ from Figure 3. This saving is achieved in the following way. We modify the protocol $\texttt{zkVC}_n$ so that for $n = 2$ it no longer performs the reduction emitting elements L and R, instead it immediately produces a proof that the commitment Y is a linear combination of three orthogonal elements, namely, X_0, X_1, H, with known to $\mathcal{P}$ coefficients.

A proof that a commitment is a linear combination of three orthogonal generators can be constructed in exactly the same way as the proof zk2ElemComm in Figure 2 for two orthogonal generators. It would take one element in $\mathbb{G}$ as the first message, and three scalars in $\mathbb{F}_{\bar{\mathsf{p}}}$ as the reply. In sum, its size would be 4, instead of 3 for zk2ElemComm. Thus, the size of the $\mathtt{zkVC}_n$ proof for $n = 2$ would be 4, instead of $2 + 3 = 5$.

We denote such an optimized $\mathtt{zkVC}_n$ as $\mathtt{zkVC}_n^{\mathbf{opt1}}$. Theorem 2 remains valid for it, because its proof is symmetrically transferred to $\mathtt{zkVC}_n^{\mathbf{opt1}}$, just the transition to a custom Schnorr-like protocol occurs at $n = 2$ instead of $n = 1$. The $\mathtt{zkVC}_n^{\mathbf{opt1}}$ protocol size is $2\log_2(n) + 2$.

6.4 SAVING ONE MORE ITEM IN ALL LOG-SIZE SCHEMES

The idea of this optimization is that, as we may have already noticed, for any $n \geqslant 1$ it is always possible to construct a custom Schnorr-like protocol for n orthogonal generators, which is HVZK and has WEE, and is of size $n + 1$. In this protocol, n scalars are transmitted as a reply, by which the orthogonal generators are then multiplied. However, it is not necessary to transmit these n scalars, only a proof of their knowledge would suffice. Moreover, this proof does not have to be HVZK, an argument having WEE only will suffice.

Speaking formally, for the first, we take the following vector commitment relation, which is actually the relation (5) with the items renamed and, also, is the relation (3) with the blinding generator H moved to the vector $\mathbf{X}$.

$$\mathcal{R} = \{\, \mathbf{X} \in \mathbb{G}^{n*},\, Y \in \mathbb{G};\, \mathbf{x} \in \mathbb{F}_{\bar{\mathsf{p}}}^{n} \mid Y = \langle \mathbf{x}, \mathbf{X} \rangle \,\}\,, \tag{50}$$

and define the following Schnorr-like protocol for it.

$\boxed{\mathtt{zkNElemComm}_n(\mathbf{X}, Y;\, \mathbf{x})}$

Relation $\mathcal{R} = \{\, \mathbf{X} \in \mathbb{G}^{n*},\, Y \in \mathbb{G};\, \mathbf{x} \in \mathbb{F}_{\bar{\mathsf{p}}}^{n} \mid Y = \langle \mathbf{x}, \mathbf{X} \rangle \,\}$ // (50)

// X in $\mathcal{R}$ satisfies $\mathrm{ort}(X)$.

$\mathcal{P}$'s input : $(\mathbf{X}, Y; \mathbf{x})$

$\mathcal{V}$'s input : $(\mathbf{X}, Y)$

$\mathcal{P}$'s output : none

$\mathcal{V}$'s output: *Accept* or *Reject*

$\boxed{\mathcal{P}}$: $\boldsymbol{\phi} \leftarrow\!\$\ \mathbb{F}_{\bar{\mathsf{p}}}^{n*}$ and computes $T = \langle \boldsymbol{\phi}, \mathbf{X} \rangle$

$\boxed{\mathcal{P} \to \mathcal{V}}$: T

$\boxed{\mathcal{V}}$: $c \leftarrow\!\$\ \mathbb{F}_{\bar{\mathsf{p}}}^{*}$

$\boxed{\mathcal{V} \to \mathcal{P}}$: c

$\boxed{\mathcal{P}}$: computes $\boldsymbol{\tau} = \boldsymbol{\phi} - c\mathbf{x}$

$\boxed{\mathcal{P} \to \mathcal{V}}$: $\boldsymbol{\tau}$

$\boxed{\mathcal{V}}$: **return**s *Accept* iff the following holds

$$T - cY \stackrel{?}{=} \langle \boldsymbol{\tau}, \mathbf{X} \rangle$$

Figure 23: Zero-knowledge argument for n element commitment relation

Properties of the protocol $\mathtt{zkNElemComm}_n$ in Figure 23 are specified in the next theorem. Note that, for $n = 2$, $\mathtt{zkNElemComm}_2$ is equivalent to zk2ElemComm in Figure 2.

Theorem 14:
For $n \in \mathbb{N}^*$, for a vector of non-zero elements $\mathbf{X} \in \mathbb{G}^{n*}$ such that all elements in $\mathbf{X}$ are orthogonal to each other, for an element $Y \in \mathbb{G}$, the protocol $\mathtt{zkNElemComm}_n$ in Figure 23 is a complete, HVZK argument having WEE for the relation (50).

Proof: The design of the protocol in Figure 23 is clearly Schnorr-like. Hence, its completeness, HVZK, and WEE can be proved in the standard way, so we do not include a detailed proof here, clarifications are the same as for zk2ElemComm in Appendix A.

For the second, in Figure 24 we define a log-size vector commitment argument $\mathtt{argVC}_n$ for the same relation (50). Note, we do use the blinding generator H neither in $\mathtt{zkNElemComm}_n$ nor in $\mathtt{argVC}_n$. Also, note that $\mathtt{zkNElemComm}_n$ is HVZK, whereas $\mathtt{argVC}_n$ is not. Its properties are specified in the following theorem.

Theorem 15:
For $n \in \mathbb{N}^*$ such that n is a power of 2, for a vector of non-zero elements $\mathbf{X} \in \mathbb{G}^{n*}$ such that all elements in $\mathbf{X}$ are orthogonal to each other, for an element $Y \in \mathbb{G}$, the protocol $\texttt{argVC}_n$ in Figure 24 is a complete argument having WEE for the relation (50).

Proof: For $n > 4$, the protocol in Figure 24 comprises the reductions from the inner product argument [2] with $\mathbf{b} = \{0\}^n$ and, hence, it is complete and has WEE for these reductions. For $n \leqslant 4$, $\mathcal{P}$ simply opens the witness to $\mathcal{V}$ and the latter checks the relation. Thus, for $n \geqslant 1$, the protocol is complete and has WEE.

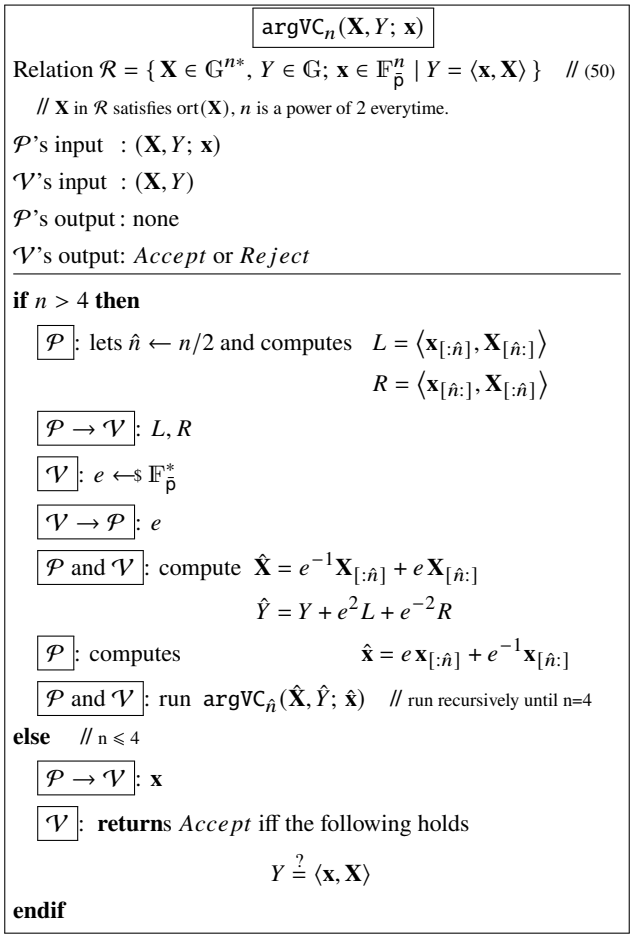

Figure 24: Efficient argument for vector commitment

Third, we combine $\texttt{zkNElemComm}_n$ with $\texttt{argVC}_n$ into a single one, as follows.

$\boxed{\mathtt{zkVC}_n^{\mathbf{opt2}}(\mathbf{X}, H, Y;\ \mathbf{a}, \alpha)}$

Relation $\mathcal{R} = \{\ \mathbf{X} \in \mathbb{G}^{n*}, H \in \mathbb{G}^*, Y \in \mathbb{G};\ \mathbf{a} \in \mathbb{F}_{\mathfrak{p}}^n, \alpha \in \mathbb{F}_{\mathfrak{p}} \mid Y = \langle \mathbf{a}, \mathbf{X} \rangle + \alpha H\ \}$ // (3)

// $\mathbf{X}, H$ in $\mathcal{R}$ satisfy $\mathrm{ort}(\mathbf{X} \cup \{H\})$, and also $(n+1)$ is a power of 2 everytime.

$\mathcal{P}$'s input : $(\mathbf{X}, H, Y; \mathbf{a}, \alpha)$

$\mathcal{V}$'s input : $(\mathbf{X}, H, Y)$

$\mathcal{P}$'s output : none

$\mathcal{V}$'s output: *Accept* or *Reject*

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: let $\hat{\mathbf{X}} \leftarrow [\mathbf{X}, H]$

$\boxed{\mathcal{P}}$: $\boldsymbol{\phi} \leftarrow\$\ \mathbb{F}_{\mathfrak{p}}^{(n+1)*}$, lets $\hat{\mathbf{x}} \leftarrow [\mathbf{x}, \alpha]$, and computes $T = \langle \boldsymbol{\phi}, \hat{\mathbf{X}} \rangle$

$\boxed{\mathcal{P} \to \mathcal{V}}$: T

$\boxed{\mathcal{V}}$: $c \leftarrow\$\ \mathbb{F}_{\mathfrak{p}}^*$

$\boxed{\mathcal{V} \to \mathcal{P}}$: c

$\boxed{\mathcal{P}}$: computes $\boldsymbol{\tau} = \boldsymbol{\phi} - c\hat{\mathbf{x}}$

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: run $\mathtt{argVC}_{n+1}(\hat{\mathbf{X}},\ T - cY;\ \boldsymbol{\tau})$

Figure 25: Efficient zero-knowledge argument for vector commitment

Theorem 16:
For a non-zero element $H \in \mathbb{G}^*$, for $n \in \mathbb{N}^*$ such that $(n+1)$ is a power of 2, for a vector of non-zero elements $\mathbf{X} \in \mathbb{G}^{n*}$ such that all elements in $\mathbf{X} \cup \{H\}$ are orthogonal to each other, for an element $Y \in \mathbb{G}$, the protocol $\mathtt{zkVC}_n^{\mathbf{opt2}}$ in Figure 25 is a complete, HVZK argument having WEE for the relation (3).

Proof: Completeness is by design. The $\mathtt{argVC}_{n+1}$ call in the last step of $\mathtt{zkVC}_n^{\mathbf{opt2}}$ has WEE by Theorem 15. Having extracted the witness $\boldsymbol{\tau}$ from it, the protocol turns out to be $\mathtt{zkNElemComm}_{n+1}$, which has WEE by Theorem 14. Thus, $\mathtt{zkVC}_n^{\mathbf{opt2}}$ has WEE. Even with the opened $\boldsymbol{\tau}$ the protocol remains HVZK by Theorem 14, so partially hiding it inside $\mathtt{argVC}_{n+1}$ doesn't make $\mathtt{zkVC}_n^{\mathbf{opt2}}$ less zero-knowledge. Thus, $\mathtt{zkVC}_n^{\mathbf{opt2}}$ is HVZK.

Note that the magic of packing the scalars $\boldsymbol{\tau}$ into the vector commitment argument is similar to the one in the work of Tsz Hon Yuen et al. [15], however, they are different. We pack the reply that looks random, whereas in [15] the true randomness sampled by prover is packed. In connection with this, providing the argument for such the scalars is sufficient in our case. Whereas cases as in [15], in our view, may either imply a weaker security model or require an additional proof that the scalars are indeed random, otherwise prover may tamper with them, more on this in Appendix Q.

As a result, $\mathtt{zkVC}_n^{\mathbf{opt2}}$ size is $2\log_2(n+1) + 1$. After substituting it for $\mathtt{zkVC}_n$, the new sizes of the Multratug and EFLRSL schemes are shown in Table 4. The scheme verification times do not change much, so we do not recalculate them. One more change is that from now we require $(n+l+1)$ and $(n+1)$ to be powers of 2, respectively.

Table 4: Optimized characteristics of the **Multratug** and **EFLRSL** schemes

	Size	Verification complexity
Multratug	$2\log_2(n+l+1) + 6l + 4$	$\boldsymbol{mexp}(\ 4n + 7l + \ldots\ ) + (n+l+2)\mathbf{H_{pt}}$
EFLRSL	$2\log_2(n+1) + 3l + 1$	$\boldsymbol{mexp}\ (\ 3n + 2l + \ldots\ ) + (n+1)\mathbf{H_{pt}}$

. . . Insignificant summands are omitted.

6.5 CHANGING KEY IMAGE TO LINEAR FORM

In some applications it is more convenient to have key image in a linear form by its secret key. For the Multratug key image $\mathcal{H}_{\mathbf{point}}(P)/x$, we can move x from the denominator to the numerator, thus turning the key image into the linear form $x\mathcal{H}_{\mathbf{point}}(P)$. This form is used, e.g., in the LSAG [7], CLSAG [4], CryptoNote [13] schemes. Our idea of x's movement is quite simple and does not require any new steps in the proofs, only an appropriate modification of them. Therefore we provide only a sketch along with an informal proof of this idea.

The main property we use is that in Multratug, according to Figure 21, due to the Lin2-2Choice lemma, each Z_k is represented as a sum of an element from the ring's first part $\mathbf{X}$ and k-th element from the second part $\mathbf{V}$. Of course, to the accuracy of the blinding component

$$Z_k = p_k X_{s_k} + p_k V_k = p_k(P_{s_k} - K + \zeta U_{s_k} - \omega A_{s_k}) + p_k(K + \omega A_k^{\mathbf{tmp}} + \chi \hat{U}_k)\,. \tag{51}$$

Inserting the value of Z_k from Figure 21 and reducing K, the equality (51) rewrites as

$$G + \zeta I_k + \chi J_k = p_k(P_{s_k} + \zeta U_{s_k} - \omega A_{s_k}) + p_k(\omega A_k^{\mathbf{tmp}} + \chi \hat{U}_k)\,. \tag{52}$$

From the equality (52), due to the random weights ζ, χ, ω and due to the fact that $\hat{U}_k$ is orthogonal to everything in the right part of the equality, verifier is convinced that, to the accuracy of blinding component, there hold

$$\begin{cases} G = p_k P_{s_k} & \text{(53a)} \\ I_k = p_k U_{s_k} & \text{(53b)} \\ J_k = p_k \hat{U}_k & \text{(53c)} \\ A_{s_k} = A_k^{\mathbf{tmp}}\ . & \text{(53d)} \end{cases}$$

The first step of our idea is that the summand $\omega A_k^{\mathbf{tmp}}$ in the element V_k expands to $\omega A_k^{\mathbf{tmp}} - \zeta U_k^{\mathbf{tmp}}$, where $U_k^{\mathbf{tmp}}$, by analogy with $A_k^{\mathbf{tmp}}$, is a blinded U_{s_k} and is included in the pre-image of $\hat{U}_k$. At the same time, we zero out I_k in Z_k. Thus, the equality (52) becomes

$$G + \chi J_k = p_k(P_{s_k} + \zeta U_{s_k} - \omega A_{s_k}) + p_k(\omega A_k^{\mathbf{tmp}} - \zeta U_k^{\mathbf{tmp}} + \chi \hat{U}_k)\,, \tag{54}$$

from which verifier is convinced that, to the accuracy of blinding component, there hold

$$\begin{cases} G = p_k P_{s_k} & \text{(55a)} \\ U_{s_k} = U_k^{\mathbf{tmp}} & \text{(55b)} \\ J_k = p_k \hat{U}_k & \text{(55c)} \\ A_{s_k} = A_k^{\mathbf{tmp}}\ . & \text{(55d)} \end{cases}$$

As the result of this step, the equality (53b) is removed, and instead the equality (55b) is added to what the verifier is convinced of.

The second step of our idea is that we expand the summand $\omega A_k^{\mathbf{tmp}} - \zeta U_k^{\mathbf{tmp}}$ again, now to $\omega A_k^{\mathbf{tmp}} - \zeta U_k^{\mathbf{tmp}} + \theta \hat{I}_k$, where $\hat{I}_k = p_k^{-1} \mathcal{H}_{\mathbf{point}}(P_{s_k})$ and θ is another randomness. The element $\hat{I}_k$, as we can see, recalling $p_k = x_k^{-1}$, is the desired key image $x_k \mathcal{H}_{\mathbf{point}}(P_{s_k})$. We create it instead of and at the same moment as I_k, which is no longer used in the protocol after the previous step. This also ensures that $\hat{I}_k$ is not blinded, since it is created before H and gets into the latter's pre-image. Finally, we add $\theta U_k^{\mathbf{tmp}}$ to Z_k, and thus the equality (54) becomes

$$G + \theta U_k^{\mathbf{tmp}} + \chi J_k = p_k(P_{s_k} + \zeta U_{s_k} - \omega A_{s_k}) + p_k(\omega A_k^{\mathbf{tmp}} - \zeta U_k^{\mathbf{tmp}} + \theta \hat{I}_k + \chi \hat{U}_k)\,, \tag{56}$$

from which verifier is convinced that, to the accuracy of blinding component, there hold

$$\begin{cases} G = p_k P_{s_k} & \text{(57a)} \\ U_{s_k} = U_k^{\mathbf{tmp}} & \text{(57b)} \\ U_{s_k} = p_k \hat{I}_k & \text{(57c)} \\ J_k = p_k \hat{U}_k & \text{(57d)} \\ A_{s_k} = A_k^{\mathbf{tmp}}\ . & \text{(57e)} \end{cases}$$

That's all. From the equality (57c) verifier is convinced that the key image $\hat{I}_k$ has the desired form, for each $k \in [0 \ldots l-1]$. The transition to this form of key image has been performed at cost of l elements in the transcript. Namely, we have added only l elements $U_k^{\mathbf{tmp}}$, $k \in [0 \ldots l-1]$ to the signature. The key images I_k's have become $\hat{I}_k$'s. Complete protocol is shown in Figure 26. A few more notes on this are in Appendix P.

$\boxed{\texttt{EFLRSLWBLI.SignAndVerify}_{l,n}(\mathsf{M}, \mathbf{P}, \mathbf{A}, A^{\mathbf{sum}}, D; \mathbf{s}, \mathbf{x}, d^{\mathbf{\Delta sum}})}$

$\mathcal{P}$'s input : $(\mathsf{M} \in \{0,1\}^\star, \mathbf{P} \in \mathbb{G}^{n*}, \mathbf{A} \in \mathbb{G}^n, A^{\mathbf{sum}} \in \mathbb{G}, D \in \mathbb{G}^*; \mathbf{s} \in [0 \ldots n-1]^l, \mathbf{x} \in \mathbb{F}_{\bar{\mathsf{p}}}^{l*}, d^{\mathbf{\Delta sum}} \in \mathbb{F}_{\bar{\mathsf{p}}})$

$\mathcal{V}$'s input : $(\mathsf{M} \in \{0,1\}^\star, \mathbf{P} \in \mathbb{G}^{n*}, \mathbf{A} \in \mathbb{G}^n, A^{\mathbf{sum}} \in \mathbb{G}, D \in \mathbb{G}^*)$

$\mathcal{P}$'s output : *Signature* // signature is a list of all $\mathcal{P} \to \mathcal{V}$ messages from this and nested protocols

$\mathcal{V}$'s output: *Accept* or *Reject*

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: **assert** all elements in $\mathbf{P}$ are non-zero and different

let $\mathbf{U} \leftarrow \{\mathcal{H}_{\mathbf{point}}(\mathbf{P}_{[i]})\}_{i=0}^{n-1}$

$\boxed{\mathcal{P}}$: allocates $\hat{\mathbf{I}} \in \mathbb{G}^{l*}$, $\mathbf{p} \in \mathbb{F}_{\bar{\mathsf{p}}}^{l*}$,

initializes **foreach** $k \in [0 \ldots l-1]$

assert $\mathbf{x}_{[k]} \neq 0$

$\mathbf{p}_{[k]} = \mathbf{x}_{[k]}^{-1}$

lets $(s_k, p_k) \leftarrow (\mathbf{s}_{[k]}, \mathbf{p}_{[k]})$,

$\hat{\mathbf{I}}_{[k]} = x_k \, \mathbf{U}_{[s_k]}$ // vector $\hat{\mathbf{I}}$ is filled in here

endforeach

$\boxed{\mathcal{P} \to \mathcal{V}}$: $\hat{\mathbf{I}}$

$\boxed{\mathcal{V}}$: **assert** all elements in $\hat{\mathbf{I}}$ are non-zero and different // $\mathcal{V}$ makes sure there is no zero I and no signer signing twice

$\epsilon \leftarrow\!\$\ \mathbb{F}_{\bar{\mathsf{p}}}^*$

$\boxed{\mathcal{V} \to \mathcal{P}}$: ϵ

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: let $H \leftarrow \mathcal{H}_{\mathbf{point}}(\epsilon)$ // thus, H is orthogonal to all known so far elements, i.e. $\mathrm{ort}(H, G, \mathbf{P}, \mathbf{A}, \mathbf{U}, \hat{\mathbf{I}}, A^{\mathbf{sum}}, D)$

$\boxed{\mathcal{P}}$: $\boldsymbol{\mu}, \hat{\boldsymbol{\mu}}, \boldsymbol{\upsilon} \leftarrow\!\$\ \mathbb{F}_{\bar{\mathsf{p}}}^{l*}$, allocates $\mathbf{A}^{\mathbf{tmp}}, \mathbf{U}^{\mathbf{tmp}} \in \mathbb{G}^{l*}$, $\boldsymbol{\alpha}, \hat{\boldsymbol{\alpha}} \in \mathbb{F}_{\bar{\mathsf{p}}}^{l*}$,

initializes **foreach** $k \in [0 \ldots l-1]$

lets $(\mu_k, \hat{\mu}_k) \leftarrow (\boldsymbol{\mu}_{[k]}, \hat{\boldsymbol{\mu}}_{[k]})$,

$\mathbf{A}_{[k]}^{\mathbf{tmp}} = \mathbf{A}_{[s_k]} + \mu_k H$ // $\mathbf{A}^{\mathbf{tmp}}$ is filled, amounts get double blinded (with D and with H)

$\boldsymbol{\alpha}_{[k]} = p_k \, \mu_k$ // $\boldsymbol{\alpha}$ is initialized here, it contains reduced $\mathbf{A}^{\mathbf{tmp}}$'s second blinding factors

$\mathbf{U}_{[k]}^{\mathbf{tmp}} = \mathbf{U}_{[s_k]} + \hat{\mu}_k H$ // $\mathbf{U}^{\mathbf{tmp}}$ is filled, U's get blinded with H

$\hat{\boldsymbol{\alpha}}_{[k]} = p_k \, \hat{\mu}_k$ // $\hat{\alpha}$ is initialized here, it contains reduced $\mathbf{U}^{\mathbf{tmp}}$'s blinding factors

endforeach

$\boxed{\mathcal{P} \to \mathcal{V}}$: $\mathbf{A}^{\mathbf{tmp}}$, $\mathbf{U}^{\mathbf{tmp}}$

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: let $\hat{\mathbf{U}} \leftarrow \{\, \mathcal{H}_{\mathbf{point}}(H, \mathbf{U}^{\mathbf{tmp}}, \mathbf{A}_{[k]}^{\mathbf{tmp}}) \,\}_{k=0}^{l-1}$

$\boxed{\mathcal{P}}$: lets $\mathbf{J} \leftarrow \{p_k \hat{\mathbf{U}}_{[k]} + \upsilon_k H\}_{k=0}^{l-1}$ // vector $\mathbf{J}$ is initialized here, it contains 'pseudo key images' built using $\hat{\mathbf{U}}$

$\boxed{\mathcal{P} \to \mathcal{V}}$: $\mathbf{J}$

$\boxed{\mathcal{V}}$: **assert** all elements in $\mathbf{A}^{\mathbf{tmp}}, \mathbf{J}$ are non-zero and different // $\mathcal{V}$ makes sure $\hat{\mathbf{U}}$ is orthogonal and there is no zero J

$\hat{\epsilon}, \zeta, \omega, \chi, \theta \leftarrow\!\$\ \mathbb{F}_{\bar{\mathsf{p}}}^*$

$\boxed{\mathcal{V} \to \mathcal{P}}$: $\hat{\epsilon}, \zeta, \omega, \chi, \theta$

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: let $K \leftarrow \mathcal{H}_{\mathbf{point}}(\hat{\epsilon})$ // thus, $\mathrm{ort}(K, H, G, \mathbf{P}, \mathbf{A}, \mathbf{U}, \mathbf{I}, A^{\mathbf{sum}}, \mathbf{A}^{\mathbf{tmp}}, \hat{\mathbf{U}}, \mathbf{J})$ holds

allocate $\mathbf{X} \in \mathbb{G}^{n*}$, $\mathbf{V}, \mathbf{Z} \in \mathbb{G}^{l*}$, $S \in \mathbb{G}$,

assign $\mathbf{X} = \mathbf{P} - \{K\}^n + \zeta \mathbf{U} - \omega \mathbf{A}$, $\quad \mathbf{V} = \{K\}^l + \omega \mathbf{A}^{\mathbf{tmp}} - \zeta \mathbf{U}^{\mathbf{tmp}} + \theta \hat{\mathbf{I}} + \chi \hat{\mathbf{U}}$,

$\mathbf{Z} = \{G\}^l + \theta \mathbf{U}^{\mathbf{tmp}} + \chi \mathbf{J}$

assign $S = A^{\mathbf{sum}} - \sum_{k=0}^{l-1} \mathbf{A}_{[k]}^{\mathbf{tmp}}$

run $\texttt{zk2ElemComm}(D, H, S;\ d^{\mathbf{\Delta sum}}, -\sum_{k=0}^{l-1} \mu_k)$

run $\texttt{zkLin22Choice}_{l,n,l}(\mathbf{X}, \mathbf{G}_{[:n]}, \mathbf{V}, \mathbf{G}_{[n:(n+l)]}, H, \mathbf{Z};\ \mathbf{s}, \mathbf{p}, \mathbf{p},\ -\omega \boldsymbol{\alpha} + \zeta \hat{\boldsymbol{\alpha}} + \theta \hat{\boldsymbol{\mu}} + \chi \boldsymbol{\upsilon})$

Figure 26: Multratug with $\hat{I} = x\mathcal{H}_{\mathbf{point}}(P)$ signing and verification

7 COMBINATION, APPLICATIONS, AND COMPARISON

7.1 COMBINING WITH OTHER PROOFS

Multratug is rooted in a single vector commitment argument and doesn't depend on the realization of the argument. Hence, Multratug can be combined with any other argument that provides a proof of vector commitment, e.g. with the inner product argument implemented according to [2] or [3].

For instance, Multratug can be combined with single or aggregate range proofs from [3], and they will share the component

$$\sum_{j=0}^{\log_2(n+l+n^{\mathbf{rangeproof}})-1} (e_j^2 L_j + e_j^{-2} R_j),$$

where $n^{\mathbf{rangeproof}}$ is equal to bitsize of the range times number of proofs aggregated.

7.2 SIGNATURE IN BLOCKCHAIN

Suppose, the Multratug signature is used to sign a transactions in an UTXO blockchain like [8, 13], where, suppose, public keys, hidden amounts, hash functions, and predefined generators follow the rules in Figures 1, 8, 19, 20.

For every transaction, its sender $\mathcal{P}$ does the following

- picks from the ledger n pairs of the form (P, A), which become transaction inputs, and makes a ring (20) of them,
- generates and places into the transaction m pairs of the form (P, A), which become the transaction outputs, for convenience considering the m hidden amounts A of these outputs as vector $\mathbf{A}^{\mathbf{out}}$,
- lets $A^{\mathbf{sum}} = \sum_{k=0}^{m-1} A_k^{\mathbf{out}}$,
- signs the transaction with the Multratug signature, knowing the vector $\mathbf{s}$ of actual signing indices at which it knows private keys,
- proves ranges for all elements in $\mathbf{A}^{\mathbf{out}}$, for example with the aggregate range proof from [3], which is easily combined with Multratug, as pointed out in Section 7.1,
- proves that each $A_k^{\mathbf{out}} \in \mathbf{A}^{\mathbf{out}}$ has the decomposition (22) with known to $\mathcal{P}$ coefficients. By the way, if a range proof protocol from [2, 3] is used for the elements of $\mathbf{A}^{\mathbf{out}}$, then proofs of $A_k^{\mathbf{out}}$'s decompositions (22) are included by that.

Thus, the transaction contains proofs that each output hidden amount $A_k^{\mathbf{out}}$ has the form (22). Also, the transaction contains Multratug, which proves that $\sum_{k=0}^{m-1} A_k^{\mathbf{out}}$ is equal to the sum $\sum_{k=0}^{l-1} A_{s_k}$ of all hidden amounts related to the signing indices $\mathbf{s}$ to the accuracy of D. Taking into account that all A_{s_k}'s in the ring are already proven having the form (22), from these proofs follows that the sum of amounts related to the actual signing keys is equal to the sum of the output amounts

$$\sum_{k=0}^{l-1} b_{s_k} = \sum_{k=0}^{m-1} b_k^{\mathbf{out}}.$$

Finally, the same Multratug proves that $\mathcal{P}$ knows private keys of signing public keys corresponding to the signing indices $\mathbf{s}$, and provides the key image vector $\mathbf{I}$ which excludes reuse of these public keys as signing keys in other transactions.

7.3 SIMPLE RING SIGNATURE

EFLRSL is a simple linkable threshold ring signature. Consequently, it can be used to implement electronic voting systems or for whistleblowing, like those in [7]. As for a not-linkable version of EFLRSL, which we do not provide in this paper, it can be easily implemented by blinding the key images, that would just require creating them after defining the blinding generator H and adding a random H-components to them.

7.4 COMPARISON WITH EXISTING SCHEMES

At present, quite a large number of log-size ring signature schemes and also signatures with balance proofs for the blockchains have already been proposed. Now we will compare our optimized Multratug and EFLRSL schemes (Table 4) with the best performing ones, namely with Omniring [6], RingCT3.0 [14], Triptych [9], and DualRing-EC [15], taking linear-size CLSAG [4] as the base line. Of course, we compare only with DDH-based, trusted-setup-free schemes without bilinear pairings.

We distinguish two gradations of scheme anonymity inherently related to the key image (linking tag) forms used. In general, if a scheme has key image or another public element of the form $x^{-1}U$, then it has lower anonymity unless there is a restriction applied on the keys. The key image forms $x^{-1}\mathcal{H}_{\mathbf{point}}(P)$ and $x\mathcal{H}_{\mathbf{point}}(P)$ do not impose any restrictions on the keys used, as they reveal no information about them. However, it is still required that the scheme has no other elements of the form $x^{-1}U$. More on this in Appendix R.

7.4.1 FOR MULTRATUG

In Table 5 we compare the schemes with the balance proofs. We denote as $\mathbf{H}_{\mathbf{sc}}$ the time of taking a hash, it is omitted when its multiplier is logarithmic or less, $\mathbf{H}_{\mathbf{pt}}$ the time of taking a hash to curve, $\boldsymbol{mexp}(N)$ the time of multi-exponentition of N summands. The schemes with 'Any keys=Yes' receive arbitrary keys, those with 'Any keys=No' remain secure only with a special key format, e.g. as in [13]. So of course, the schemes with key image forms $x^{-1}\mathcal{H}_{\mathbf{point}}(P)$ and $x\mathcal{H}_{\mathbf{point}}(P)$ have an additional summand of roughly $n\mathbf{H}_{\mathbf{pt}}$ in their verification complexity formulas. We do not know whether the Omniring version with the key image $x\mathcal{H}_{\mathbf{point}}(P)$ accepts arbitrary keys, as this version of Omniring is presented only in connection with an integration to a CryptoNote [13] based system in the corresponding paper [6].

We exclude key images together with input/output accounts, which occupy the same space for all schemes. Also, we do not include the output range proofs, assuming they are separated into distinct units, although by Section 7.1 our scheme effectively integrates with them, as does Omniring [6]. Batch verification time, for our scheme explained in Section 6.2, is generally 25%. . . 50% less for all log-size schemes due to common generators merging, we do not show it.

According to Table 5, assuming ring size is large, say $n = 2^{10}$, and the number of inputs is very limited, say, $l < 5$ with a bias toward $l = 2$ which is in line with [14, 6, 9], Multratug provides the shortest proof size for $l = 1$. For $l = 2$ our proof size is almost equal to size of Omniring. For bigger l Omniring provides the shortest proof size. It is to be noted that the verification time of Multratug grows noticeably slower than Omniring's time, since Multratug's actual decoy set grows as $n + l$ whereas Omniring's one grows as $n\,l$, which can be seen from the expressions under the $\log_2$.

Table 5: Comparison of LRS schemes that simultaneously prove the balance

	Size	Verification complexity	Key image	Any keys
CLSAG*	$n + 2$	$(n+2)\mathbf{H}_{\mathbf{sc}} + 2n\,\boldsymbol{mexp}(3) + n\mathbf{H}_{\mathbf{pt}}$	$x\,\mathcal{H}_{\mathbf{point}}(P)$	Yes
Triptych*	$3\log_2(n) + 8$	$\boldsymbol{mexp}(\,2n + \ldots\,)$	$x^{-1}U$	No
RingCT3.0	$2\log_2(n\,l) + l + 17$	$\boldsymbol{mexp}(\,2\,n\,l + \ldots\,) + \boldsymbol{mexp}(l+1) + \ldots$	$x^{-1}U$	No
Omniring	$2\log_2(n\,l + n + 3l + 3) + 9$	****	$x^{-1}U$	No
Omniring	$2\log_2(n\,l + n + 3l + 3) + 9$	****	$x\,\mathcal{H}_{\mathbf{point}}(P)$	Probably**
Multratug	$2\log_2(n + l + 1) + 5l + 4$	$\boldsymbol{mexp}(\,4n + 7l + \ldots\,) + (n + l + 2)\mathbf{H}_{\mathbf{pt}}$	$x^{-1}\mathcal{H}_{\mathbf{point}}(P)$	Yes
Multratug**	$2\log_2(n + l + 1) + 6l + 4$	$\boldsymbol{mexp}(\,4n + 8l + \ldots\,) + (n + l + 2)\mathbf{H}_{\mathbf{pt}}$	$x\,\mathcal{H}_{\mathbf{point}}(P)$	Yes

* Authors did not specify any optimized threshold version, assuming it takes up l times the size.
** Authors provide security model only for the less secure key image form $x^{-1}U$.
*** Scheme version with linear linking tag, Section 6.5.
**** Authors did not specify formula, we assume the quantity is average in its class.
. . . Insignificant summands are omitted.

7.4.2 FOR EFLRSL

In Table 6 we compare the streamlined versions of the schemes, which are ring signatures with one actual signer. So, we take our EFLRSL signature for $l = 1$. We also include in the comparison the DualRing-EC [15] signature, which is published as the shortest known so far. For this comparison, we don't distinguish between the regular ring signatures and the linkable ones. When both versions are available we take the regular one, in this case the linkable version usually takes up one more element of space.

According to Table 6, for a large ring size, such that $\log_2(n + 1) \approx \log_2(n)$ with indistinguishable difference, both the DualRing-EC and EFLRSL signatures have the shortest size. However, EFLRSL has a stronger security model, which is explained in Appendix Q. Thus, it appears that the EFLRSL signature for $l = 1$ is the shortest one known to date for environments in which malformed keys are permitted.

Table 6: Comparison of DL-based ring signatures

	Size	Verification complexity
CLSAG	$n+1$	$n\,\mathbf{H_{sc}} + n\,\boldsymbol{mexp}(2)$
RingCT3.0	$2\log_2(n)+14$	$\boldsymbol{mexp}(\,2n+\ldots\,)+\ldots$
Omniring	$2\log_2(n+2)+9$	***
EFLRSL*	$2\log_2(n+1)+4$	$\boldsymbol{mexp}(\,3n+\ldots\,)+(n+1)\mathbf{H_{pt}}$
DualRing-EC**	$2\log_2(n)+4$	$\boldsymbol{mexp}(\,n+\ldots\,)$

* Only linkable version of the ring signature is available.
** See comments in Appendix Q.
*** Authors did not specify formula, we assume the quantity is average in its class.
. . . Insignificant summands are omitted.

8 CONCLUSION

In this paper we have created a trusted-setup-free, pairings-free, DDH-based linkable threshold ring signature called Multratug, which simultaneously provides a proof of the balance. We have shown Multratug can be used to sign transactions with hidden amounts in blockchain. Built on top of a log-size vector commitment argument, it can be combined with other proofs, e.g. with the log-size range proofs. We have also created a version of Multratug with a linear by private key linking tag.

A lightweight version of Multratug, called EFLRSL, which we have designed along the way and which does not involve account balances at all, can be used as a mere linkable ring signature with threshold or without it.

In comparison with the recent schemes that deliver the same, it turns out that Multratug and EFLRSL are on a par with the most efficient of them, and for several rather typical cases outperform. The comparison data are summarized in Tables 5, 6. Multratug and EFLRSL have a strong security model that permits malformed keys in their rings. Also, they employ a form of linking tag that ensures anonymity and unforgeability even if an adversarially chosen distribution of keys is used. Thus, Multratug and EFLRSL are suitable for use in real systems with high security requirements.

While constructing our schemes we have designed two logarithmic proofs of membership, which may be of an independent interest. Under DDH these proofs have the special honest verifier zero-knowledge and computational witness-extended emulation properties, that we prove in the Lin2-Choice and Lin2-2Choice lemmas. As both these protocols are based on an arbitrary vector commitment argument, which is only required to be honest verifier zero-knowledge and having computational witness-extended emulation, they can be further optimized.

REFERENCES

[1] Dan Boneh and Victor Shoup. *A Graduate Course in Applied Cryptography*. Dan Boneh's publications web page, `http://crypto.stanford.edu/~dabo/pubs/abstracts/bookShoup.html`. `https://toc.cryptobook.us/book.pdf`. 2020.

[2] Benedikt Bünz et al. "Bulletproofs: Short proofs for confidential transactions and more". In: *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2018, pp. 315–334.

[3] Heewon Chung et al. *Bulletproofs+: Shorter Proofs for Privacy-Enhanced Distributed Ledger*. Cryptology ePrint Archive, Report 2020/735. `https://ia.cr/2020/735`. 2020.

[4] Brandon Goodell, Sarang Noether, and RandomRun. *Concise Linkable Ring Signatures and Forgery Against Adversarial Keys*. Cryptology ePrint Archive, Report 2019/654. `https://ia.cr/2019/654`. 2019.

[5] Jens Groth and Markulf Kohlweiss. "One-out-of-many proofs: Or how to leak a secret and spend a coin". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2015, pp. 253–280.

[6] Russell WF Lai et al. "Omniring: Scaling private payments without trusted setup". In: *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. 2019, pp. 31–48.

[7] Joseph K Liu, Victor K Wei, and Duncan S Wong. "Linkable Spontaneous Anonymous Group Signature for Ad Hoc Groups (Extended Abstract)". In: *Proc. Ninth Australasian Conf. Information Security and Privacy (ACISP)*. 2004.

[8] Satoshi Nakamoto. *Bitcoin: A peer-to-peer electronic cash system*. `https://bitcoin.org/bitcoin.pdf`. 2008.

[9] Sarang Noether and Brandon Goodell. *Triptych: logarithmic-sized linkable ring signatures with applications*. Cryptology ePrint Archive, Report 2020/018. https://ia.cr/2020/018. 2020.

[10] Claus-Peter Schnorr. "Efficient Signature Generation by Smart Cards". In: *J. Cryptology* 4.3 (1991), pp. 161–174.

[11] Anton A. Sokolov. *Lin2-Xor Lemma and Log-size Linkable Threshold Ring Signature*. Cryptology ePrint Archive, Report 2020/688. https://ia.cr/2020/688. 2020.

[12] Patrick P. Tsang et al. *Separable Linkable Threshold Ring Signatures*. Cryptology ePrint Archive, Report 2004/267. https://ia.cr/2004/267. 2004.

[13] Nicolas Van Saberhagen. *CryptoNote v 2.0*. https://cryptonote.org/whitepaper.pdf. 2013.

[14] Tsz Hon Yuen et al. *RingCT 3.0 for Blockchain Confidential Transaction: Shorter Size and Stronger Security*. Tech. rep. Cryptology ePrint Archive, Report 2019/508, 2019. https://eprint.iacr.org/2019/508, 2019.

[15] Tsz Hon Yuen et al. *DualRing: Generic Construction of Ring Signatures with Efficient Instantiations*. Cryptology ePrint Archive, Paper 2021/1213. https://eprint.iacr.org/2021/1213. 2021.

A PROOF OF 2-ELEMENT COMMITMENT

Proof: [Theorem 1] The completeness, HVZK, and WEE of the protocol in Figure 2 for the relation (2) can be proved using the well-known methods. They are the methods the completeness, HVZK, and WEE of the Schnorr identification scheme [10] and other Schnorr-like protocols in [1, 3, 11] are proved. We will not repeat descriptions of these methods here to save space and refer the interested reader to the mentioned works, where they are presented in full detail.

B PROOF OF VECTOR COMMITMENT

Proof: [Theorem 2] The $\texttt{zkVC}_n$ protocol in Figure 3 is a slightly modified subset version of the Bulletproofs logarithmic inner product argument from [2]. There are three modifications to it, as follows

- The inner product argument described in [2] has no HVZK property, we append this property to it the same way this is done in [3], namely by adding a blinding component to all transmitted elements. We do not provide a proof of HVZK for our $\texttt{zkVC}_n$ protocol here; it is completely identical to the HVZK proof in [3].
- With the above modification, the $\texttt{zkVC}_n$ protocol in Figure 3 is a subset case, namely $\mathbf{b} = \mathbf{0}^n$, of the inner product argument from [2] for the relation (4). Taking into account the appended HVZK property and renaming elements, our protocol proves the relation (3).
- For the case $n = 1$ in $\texttt{zkVC}_n$ we use the custom zero-knowledge zk2ElemComm protocol, which is complete, HVZK, and has WEE by Theorem 1.

Each of the three above modifications clearly does not override the completeness and WEE properties of the Bulletproofs logarithmic inner product argument. Also, the first modification adds the HVZK property. Thus, our protocol $\texttt{zkVC}_n$ in Figure 3 is a complete, HVZK argument having WEE for the relation (3).

C PROOF OF 3-TUPLE RANDOM WEIGHTING

Proof: [Theorem 3] Completeness and HVZK properties of the zk3ElemRW protocol in Figure 4 directly follow from that zk3ElemRW adds nothing to transcript of a protocol called in the last step of it, which in its turn is complete and HVZK by the premise.

WEE property of the zk3ElemRW protocol is also easy to establish, we will not provide a detailed proof here to save space, only the following sketch.

First, note that due to orthogonality of H to all other generators, components proportional to H of all participating elements can be considered separately and be omitted in the main consideration. For the H components of the protocol, it suffices only that the factor $\hat{\alpha}$ be calculated as $\hat{\alpha} = \alpha + \delta_1\beta + \delta_2\gamma$.

Second, witness extraction can be accomplished in a well-known way, e.g., as in the proof of the RandomWeighting-WEE lemma in [11].

Third, to ascertain that the witness a has only one possible value in this protocol, we can write Z, F, E as

$$\begin{cases} Z = z_P P + z_Q Q + z_R R \\ F = f_P P + f_Q Q + f_R R \\ E = e_P P + e_Q Q + e_R R \end{cases}, \tag{58}$$

since it is clear that, when H is already excluded from the consideration, the elements Z, F, E cannot have components beyond the linear span of P, Q, R without breaking the DL assumption. Inserting the decomposition (58) into the equality $Y = aX$, we obtain

$$\operatorname{rank}\left(\begin{bmatrix} 1 & \delta_1 \text{ or } 0, \text{ if } Q = 0 & \delta_2 \text{ or } 0, \text{ if } R = 0 \\ z_P + \delta_1 f_P + \delta_2 e_P & z_Q + \delta_1 f_Q + \delta_2 e_Q & z_R + \delta_1 f_R + \delta_2 e_R \end{bmatrix}\right) < 2\,,$$

which immediately yields, for some unique a

$$\begin{cases} Z = aP \\ F = aQ \\ E = aR \end{cases},$$

and from where it can be understood why we are demanding $P \neq 0 \wedge (Q \neq 0 \vee R \neq 0)$.

D PROOF OF SIMMETRIC VECTOR COMMITMENT

Proof: [Theorem 4] The protocol $\texttt{zkSVC}_{3,n}$ in Figure 5 adds nothing to the transcript of the protocol $\texttt{zkVC}_n$ (or, to be precise, to transcript of any complete, HVZK, and WEE protocol called in the last step), thus inheriting the HVZK property from the latter. Completeness of the protocol $\texttt{zkSVC}_{3,n}$ is clear. WEE property of the protocol is easy to establish, the sketch follows.

First of all, we exclude H from all considerations for the same reason as in Appendix C. Then, because of orthogonality of all non-zero elements in $\mathbf{P} \cup \mathbf{Q} \cup \mathbf{R}$, each of the elements Z, F, and E decomposes into a weighted direct sum of $\mathbf{P}, \mathbf{Q}, \mathbf{R}$ respectively. Therefore, to prove the WEE property of $\texttt{zkSVC}_{3,n}$ it suffices to prove WEE for $\texttt{zkSVC}_{3,1}$.

In its turn, $\texttt{zkSVC}_{3,1}$ is equivalent to the protocol $\texttt{zk3ElemRW}$ in Figure 4, so by Theorem 3 $\texttt{zkSVC}_{3,1}$ has WEE. Thus we obtain WEE for $\texttt{zkSVC}_{3,n}$.

E PROOF OF LIN2-CHOICE LEMMA

Proof: [Theorem 5] Completeness and HVZK of the $\texttt{zkLin2Choice}_n$ protocol in Figure 7 are clear. We exclude H from all considerations for the same reason as in Appendix C.

Let's prove the WEE property of the protocol. In the last step of $\texttt{zkLin2Choice}_n$ there is a call to

$$\texttt{zkSVC}_{2,n}(\mathbf{P}, \mathbf{c} \circ \mathbf{Q}, H, Z, rF; \mathbf{a}, \alpha, \hat{\beta}),$$

and hence by Theorem 4 there holds the relation

$$\begin{cases} Z = \langle \mathbf{a},\, \mathbf{P} \rangle \\ rF = \langle \mathbf{a},\, \mathbf{c} \circ \mathbf{Q} \rangle \end{cases}, \tag{59}$$

where $\mathbf{a} \in \mathbb{F}_{\mathsf{p}}^n$ is extracted by the $\texttt{zkSVC}_{2,n}$ protocol extractor.

Thus, if $\mathbf{a}$ contains only one non-zero scalar, say, under index j, then the sought witness p is extracted together with the index s, namely, $p = a_j$, $s = j$. If $\mathbf{a} = \{0\}^n$ is the case, then the witness p is extracted as zero, the index s has no meaning.

Let's show that $\mathbf{a}$ cannot contain more than one non-zero scalar, otherwise the $\texttt{zkLin2Choice}_n$ protocol extractor is able to break the DL assumption. Suppose that $\mathbf{a}$ contains at least two non-zeros, a_j and a_k, under the indices j and k such that $j \neq k$. Writing out Z and rF as weighted direct sums of $\mathbf{P}$ and $\mathbf{Q}$, respectively, according to the equalities (59) we obtain that having unwound the $\texttt{zkSVC}_{2,n}$ call the extractor has $Z, F, \mathbf{c}, r, \mathbf{a}$ such that the following two equalities hold

$$Z = \sum_{i=0}^{n-1} a_i P_i\,, \tag{60}$$

$$rF = \sum_{i=0}^{n-1} a_i c_i Q_i\,, \tag{61}$$

where $r \neq 0$, otherwise the equality (61) would immediately produce a contradiction with $\operatorname{ort}(\mathbf{Q})$.

Let the extractor unwinds to the point where the challenges $\mathbf{c}$ were generated, and resumes obtaining new $\mathbf{c}', r', \mathbf{a}'$. Thus, by the equality (61) there holds $r' \neq 0$, and by the equality (60) there holds $\mathbf{a}' = \mathbf{a}$. By excluding F from the equality (61) the extractor obtains

$$0 = \sum_{i=0}^{n-1} a_i (\frac{c_i}{r} - \frac{c'_i}{r'}) Q_i \,. \tag{62}$$

Due to ort($\mathbf{Q}$) all weights of Q_i's in the equality (62) must be zero, otherwise the extractor breaks the DL assumption.

According to our supposition, $a_j \neq 0$ and $a_k \neq 0$, so we write out two equations for the weights of Q_j and Q_k

$$\begin{cases} 0 = & \frac{c_j}{r} - \frac{c'_j}{r'} \\ 0 = & \frac{c_k}{r} - \frac{c'_k}{r'} \end{cases} , \tag{63}$$

where we have already performed division by non-zero a_j and a_k. Since $r \neq 0$ and $r' \neq 0$, the system (63) reduces to

$$\frac{c_k}{c'_k} = \frac{c_j}{c'_j} \,, \tag{64}$$

which holds only with negligible probability. Therefore, if there is more than one non-zero element in $\mathbf{a}$, then the extractor with overwhelming probability obtains one or more non-zero weights of Q_i's in the equality (62). Thus, under our supposition, the extractor breaks the DL assumption by expressing Q_j through the elements of $\mathbf{Q} \setminus \{Q_j\}$, hence our supposition is incorrect.

By this we have proved that the extractor with overwhelming probability finds witness for the relation (12) and, thus, the protocol $\texttt{zkLin2Choice}_n$ has WEE.

F SIGNATURE EFLRS1

Proof: [Theorem 6] As follows from Figure 10, EFLRS1 is a linkable ring signature by definition (we imply the `EFLRS1.Link` method is defined usual way, i.e. matching key images, e.g., as in [7]).

All the listed properties of the EFLRS1 signature are proved by well-known methods, such as, for example, in [7, 4, 11], which rely on the key image of the form of $x^{\pm 1}\boldsymbol{\mathcal{H}}_{\mathbf{point}}(P)$ and on completeness, HVZK, and WEE of the underlying proving system. We do not describe these proofs here due to their volume; instead, we refer the interested reader to the cited publications.

G PROOF OF MULTIPLE VECTOR COMMITMENTS

Proof: [Theorem 7] As can be seen from Figure 12, the protocol $\texttt{zkMVC}_{l,n}$ adds nothing to the transcript of the protocol $\texttt{zkVC}_n$, thus inheriting the HVZK property. Completeness of the protocol $\texttt{zkMVC}_{l,n}$ is clear. Let's prove the protocol WEE property.

This time, to show an example, we will not exclude the generator H from our consideration. We add H to $\mathbf{X}$ obtaining the expanded vector $\bar{\mathbf{X}} \in \mathbb{G}^{n+1}$

$$\bar{\mathbf{X}} = \begin{bmatrix} \mathbf{X} \\ H \end{bmatrix} .$$

At the same time, we attach the vector of blinding factors $\boldsymbol{\alpha} \in \mathbb{F}_{\bar{\mathsf{p}}}^{l}$ to the witness matrix $\mathfrak{a} \in \mathbb{F}_{\bar{\mathsf{p}}}^{l \times n}$, and thus define the expanded witness matrix $\bar{\mathfrak{a}} \in \mathbb{F}_{\bar{\mathsf{p}}}^{l \times (n+1)}$ as

$$\bar{\mathfrak{a}} = [\mathfrak{a} \;\; \boldsymbol{\alpha}].$$

Also, we combine $\mathbf{a} \in \mathbb{F}_{\bar{\mathsf{p}}}^{n}$ with $\alpha \in \mathbb{F}_{\bar{\mathsf{p}}}$, and thus define $\bar{\mathbf{a}} \in \mathbb{F}_{\bar{\mathsf{p}}}^{n+1}$

$$\bar{\mathbf{a}} = \begin{bmatrix} \mathbf{a} \\ \alpha \end{bmatrix} .$$

Having unwound the $\texttt{zkVC}_n$ call, extractor obtains $\bar{\mathbf{a}}$. As a result, for each i-th column $\mathfrak{a}_{[:,i]}$ of the matrix $\mathfrak{a}$ there holds the equality

$$\bar{\mathbf{a}}_{[i]} = \boldsymbol{\xi}^{\intercal} \cdot \bar{\mathfrak{a}}_{[:,i]} \,. \tag{65}$$

The extractor repeats the unwinding l times with re-sampled challenges $\boldsymbol{\xi}$. This way the equality (65) repeated l times turns into a matrix equation with random matrix of size $l \times l$, from which the extractor recovers each i'th column $\bar{\mathfrak{a}}_{[:,i]}$, $i \in [0 \ldots n]$ of the matrix $\bar{\mathfrak{a}}$. Thus, the extractor recovers the sought witness $\bar{\mathfrak{a}}$.

H PROOF OF THE PROPERTIES OF MANY-OUT-OF-MANY PROOF

Proof: [Theorem 8] Completeness and HVZK of the $\texttt{zkLin2mChoice}_{n,l}$ protocol in Figure 13 are clear. Let's prove the WEE property of the protocol. We will consider H this time.

First, extractor uses the $\texttt{zkMVC}_{l,n}$ protocol extractor, which exists by Theorem 7, and restores witness $(\mathfrak{a}, \hat{\boldsymbol{\alpha}})$ from the $\texttt{zkMVC}_{l,n}$ call in the last step of $\texttt{zkLin2mChoice}_{n,l}$. After that, for every $k \in [0 \mathinner{..} l-1]$, it assigns

$$(\mathbf{a}, \hat{\alpha}) \leftarrow (\mathfrak{a}_{[k]}, \hat{\boldsymbol{\alpha}}_{[k]}) \,,$$

and proceeds with the extraction using the $\texttt{zkLin2Choice}_n$ protocol extractor, which exists by Theorem 5, as though the values of $\mathbf{a}, \hat{\alpha}$ were obtained from $\texttt{zkVC}_n$ in the last step of $\texttt{zkLin2Choice}_n$. This way the extractor obtains witness (p, α), and maps it to k-th positions in $\mathbf{p}$ and $\boldsymbol{\alpha}$, respectively.

We have shown how the extractor restores witness $(\mathbf{p}, \boldsymbol{\alpha})$ for the relation (18) and, hence, the $\texttt{zkLin2mChoice}_{n,l}$ protocol has WEE.

I SIGNATURE EFLRSL FOR L=1

As can be seen from Figure 14, for $l = 1$ the EFLRSL protocol is the same as the EFLRS1 protocol in Figure 10, with the variables and calls renamed. Although the multiplier ξ_0 is applied to both commitment and witness in the nested $\texttt{zkVC}_n$ call, this doesn't distort the correspondence. Thus, by Theorem 6, for $l = 1$, all the properties listed in Theorem 9 hold.

J SIGNATURE EFLRSL FOR L $\geqslant$ 1

Proof: [Theorem 9] The case $l = 1$ proof is provided in Appendix I.

As can be seen from Figure 14, the EFLRSL protocol is a linkable threshold ring signature by definition (we imply the `EFLRSL.Link` method is defined usual way, i.e. matching key images).

All the listed properties of the EFLRSL signature can be proved by well-known methods, for example, by assuming that any of the properties does not hold, and reducing this case to the case $l = 1$, i.e. to the contradiction with the already proven in Appendix I. In this case, as e.g. in [7, 12, 4], the key image form $x^{\pm 1}\boldsymbol{\mathcal{H}}_{\mathbf{point}}(P)$ and completeness, HVZK, and WEE of the underlying proving system are used.

We do not present the proofs here because of their volume, referring the interested reader to the cited publications.

K PROOF OF SIMPLIFIED LIN2-2CHOICE LEMMA

Proof: [Theorem 10] Completeness and HVZK properties of the $\texttt{zkLin22sChoice}_{n,m}$ protocol in Figure 16 are clear. We exclude H from the consideration for the same reason as in Appendix C.

Let's prove the protocol WEE property. In the last step of $\texttt{zkLin22sChoice}_{n,m}$ there is a call to

$$\texttt{zkSVC}_{3,n}\left(\begin{bmatrix}\mathbf{P}\\ \mathbf{V}\end{bmatrix}, \begin{bmatrix}\mathbf{c}_{[:n]} \circ \mathbf{Q}\\ \mathbf{0}^m\end{bmatrix}, \begin{bmatrix}\mathbf{0}^n\\ \mathbf{c}_{[n:]} \circ \mathbf{W}\end{bmatrix}, H, Z, rF, c_{n+t}E; \mathbf{a}, \alpha, \hat{\beta}, \hat{\gamma}\right),$$

and hence by Theorem 4 there holds the relation

$$\begin{cases} Z & = \langle \mathbf{a}_{[:n]}, \mathbf{P}\rangle + \langle \mathbf{a}_{[n:]}, \mathbf{V}\rangle \\ rF & = \langle \mathbf{a}_{[:n]}, \mathbf{c}_{[:n]} \circ \mathbf{Q}\rangle \\ c_{n+t}E & = \langle \mathbf{a}_{[n:]}, \mathbf{c}_{[n:]} \circ \mathbf{W}\rangle \end{cases}, \tag{66}$$

with the witness $\mathbf{a} \in \mathbb{F}_{\bar{\mathfrak{p}}}^{n+m}$ restored by the $\texttt{zkSVC}_{3,n}$ protocol extractor.

Due to $\mathrm{ort}(\mathbf{P}, \mathbf{V}, \mathbf{Q}, \mathbf{W})$, having $Z = Z_P + Z_V$ according to the formula (36), the system (66) breaks down into two subsystems

$$\begin{cases} Z_P & = \langle \mathbf{a}_{[:n]}, \mathbf{P}\rangle \\ rF & = \langle \mathbf{a}_{[:n]}, \mathbf{c}_{[:n]} \circ \mathbf{Q}\rangle \end{cases}, \tag{67}$$

$$\begin{cases} Z_V & = \langle \mathbf{a}_{[n:]}, \mathbf{V}\rangle \\ c_{n+t}E & = \langle \mathbf{a}_{[n:]}, \mathbf{c}_{[n:]} \circ \mathbf{W}\rangle \end{cases}. \tag{68}$$

Each of the systems (67), (68) is similar to the system (59) and, therefore, by applying the same reasons to each of them as in the proof of the WEE property of the Lin2-Choice lemma in Appendix E, we obtain the following two equations respectively

$$Z_P = pP_s \ , \tag{69}$$

$$Z_V = vV_{n+\tilde{s}} \ , \tag{70}$$

where p and v are scalars known to prover, and $s, \tilde{s}$ are indices also known to it (if $p = 0$ or $v = 0$, then respectively s or $\tilde{s}$ is undefined). Furthermore, when obtaining the equality (69) from the subsystem (67), we take r as a response to the challenges $\mathbf{c}_{[:n]}$, whereas obtaining the equality (70) from the subsystem (68), we take c_{n+t} as the response to the challenges $\mathbf{c}_{[n:]}$.

If $v \neq 0$ and $\tilde{s} \neq t$, then the extractor breaks the DL assumption by establishing a linear relationship between at least two different elements from the orthogonal set $\mathbf{R}$, hence we let $\tilde{s} = t$ for $v \neq 0$ and write the equality (70) as

$$Z_V = vV_{n+t} \ . \tag{71}$$

Now, recalling that Z decomposes into the sum $Z = Z_P + Z_V$ by the formula (36) which is discussed in Section 1.2.11, the extractor comes to the conclusion that the restored by the formulas (69), (71) values of (p, v, s) are the sought witnesses for the relation (29). Thus, we have proved the WEE property of $\texttt{zkLin22sChoice}_{n,m}$.

L PROOF OF MULTIPLE SIMMETRIC VECTOR COMMITMENTS

Proof: [Theorem 11] As can be seen from Figure 17, the $\texttt{zkMSVC}_{l,3,n}$ protocol adds nothing to the transcript of the $\texttt{zkMVC}_{l,n}$ protocol, thus inheriting the HVZK property. Completeness of the $\texttt{zkMSVC}_{l,3,n}$ protocol is clear from Figure 17. We exclude H from all considerations for the same reason as in Appendix C.

Let's prove the WEE property of the protocol. Having unwound the $\texttt{zkMVC}_{l,n}$ call, extractor obtains a matrix $\mathfrak{a} \in \mathbb{F}_{\bar{\mathsf{p}}}^{l \times n}$ such that according to the relation (17)

$$\mathbf{Y} = \mathfrak{a} \cdot \mathbf{X} \, . \tag{72}$$

Thus, for each element $Y_j = \mathbf{Y}_{[j]}$, $j \in [0 \ldots l-1]$, and for the corresponding row $\mathfrak{a}_{[j,:]}$ of the matrix $\mathfrak{a}$, there holds

$$Y_j = \mathfrak{a}_{[j,:]} \cdot \mathbf{X} \, . \tag{73}$$

At the same time, due to the equalities (73), the $\texttt{zkMVC}_{l,n}$ protocol can be viewed as l independent, except for the common challenges (δ_1, δ_2), instances of the $\texttt{zkSVC}_{3,n}$ protocol. Therefore, by Theorem 4, the restored by the extractor matrix $\mathfrak{a}$ is the sought witness.

M PROOF OF LIN2-2CHOICE LEMMA

Proof: [Theorem 12] Completeness and HVZK of the protocol $\texttt{zkLin22Choice}_{l,n,m}$ in Figure 18 are clear. Particularly, note that the vectors $\mathbf{F}$ and $\mathbf{E}$ do not reveal any information since their elements are blinded with H. We further exclude H from all considerations for the same reason as in Appendix C.

Let's prove the protocol WEE property. In the last step of $\texttt{zkLin22Choice}_{l,n,m}$ there is a call to

$$\texttt{zkMSVC}_{l,3,(n+m)} \left(\begin{bmatrix} \mathbf{P} \\ \mathbf{V} \end{bmatrix}, \begin{bmatrix} \mathbf{c}_{[:n]} \circ \mathbf{Q} \\ \mathbf{0}^m \end{bmatrix}, \begin{bmatrix} \mathbf{0}^n \\ \mathbf{c}_{[n:]} \circ \mathbf{W} \end{bmatrix}, H, \mathbf{Z}, \mathbf{r} \circ \mathbf{F}, \mathbf{c}_{[n:(n+l)]} \circ \mathbf{E};\ \mathfrak{a}, \alpha, \hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\gamma}} \right) ,$$

and hence, by Theorem 11, there holds the following system of equalities

$$\begin{cases} \mathbf{Z} & = \mathfrak{a} \cdot \begin{bmatrix} \mathbf{P} \\ \mathbf{V} \end{bmatrix} \\ \mathbf{r} \circ \mathbf{F} & = \mathfrak{a} \cdot \begin{bmatrix} \mathbf{c}_{[:n]} \circ \mathbf{Q} \\ \mathbf{0}^m \end{bmatrix} \\ \mathbf{c}_{[n:(n+l)]} \circ \mathbf{E} & = \mathfrak{a} \cdot \begin{bmatrix} \mathbf{0}^n \\ \mathbf{c}_{[n:]} \circ \mathbf{W} \end{bmatrix} \end{cases} , \tag{74}$$

where the matrix $\mathfrak{a} \in \mathbb{F}_{\bar{\mathsf{p}}}^{l \times (n+m)}$ is the witness restored by the $\texttt{zkMSVC}_{l,3,(n+m)}$ protocol extractor.

Furthermore, the system (74) is l systems of the form (66), with proper renaming, for each row $\mathfrak{a}_{[t,:]}, t \in [0 \dots l-1]$ of the matrix $\mathfrak{a}$. Namely, the system (74) is the following l systems

$$\begin{cases} Z_t & = \langle \mathfrak{a}_{[t,:n]},\, \mathbf{P} \rangle & + & \langle \mathfrak{a}_{[t,n:]},\, \mathbf{V} \rangle \\ r_t F_t & = \langle \mathfrak{a}_{[t,:n]},\, \mathbf{c}_{[:n]} \circ \mathbf{Q} \rangle & & \\ c_{n+t} E_t & = & & \langle \mathfrak{a}_{[t,n:]},\, \mathbf{c}_{[n:]} \circ \mathbf{W} \rangle \end{cases} , \tag{75}$$

for each $t \in [0 \dots l-1]$.

The $\texttt{zkLin22Choice}_{l,n,m}$ protocol in Figure 18 comprises, up to the point of calling $\texttt{zkMSVC}_{l,3,(n+m)}$ and with the appropriate renaming, l parallel instances of the protocol $\texttt{zkLin22sChoice}_{n,m}$ from Figure 16. Hence, given l parallel systems (75) for $t \in [0 \dots l-1]$, the extractor performs l times, for each t, the same calculations as in Appendix K. This way it obtains l witnesses (p_t, v_t, s_t), $t \in [0 \dots l-1]$ for l instances of the relation (29). That is, for each extracted tuple (p_t, v_t, s_t) there holds

$$Z_t = p_t P_{s_t} + v_t V_t \,, \tag{76}$$

that means witnesses for the relation (39) are found and, hence, WEE property of the $\texttt{zkLin22Choice}_{l,n,m}$ protocol is proven.

N PROOF OF CLAIM ABOUT LIN2-2CHOICE PROTOCOL CALL

Proof: [Claim 1] By Theorem 12 the call

$$\texttt{zkLin22Choice}_{l,n,l}((\mathbf{X}, \mathbf{G}_{[:n]}, \mathbf{V}, \mathbf{G}_{[n:(n+l)]}, H, \mathbf{Z}; \dots)$$

in the last step of the EFLRSLWB (Multratug) scheme in Figure 21 proves the relation (39).

Let's demonsrate that this call also proves that $\mathbf{v} = \mathbf{p}$ in the relation (39), where $\mathbf{X}, \mathbf{V}, \mathbf{Z}$ are defined according to the EFLRSLWB scheme. Writing out their definitions here

$$\begin{aligned} \mathbf{X} &= \mathbf{P} - \{K\}^n + \zeta \mathbf{U} - \omega \mathbf{A} \,, \\ \mathbf{V} &= \{K\}^l + \omega \mathbf{A}^{\mathbf{tmp}} + \chi \hat{\mathbf{U}} \,, \\ \mathbf{Z} &= \{G\}^l + \zeta \mathbf{I} + \chi \mathbf{J} \,. \end{aligned}$$

Suppose the opposite, i.e., that for some $k \in [0 \dots l-1]$ there holds $v_k \neq p_k$. Then the $\texttt{zkLin22Choice}_{l,n,m}$ protocol extractor extracts such $\mathbf{v}, \mathbf{p}$, and for some index s_k there holds, according to relation (39)

$$G + \zeta I_k + \chi J_k = p_k (P_{s_k} - K + \zeta U_{s_k} - \omega A_{s_k}) + v_k (K + \omega A_k^{\mathbf{tmp}} + \chi \hat{U}_k) \,. \tag{77}$$

Note that we omit writting out the H component for the same reason as in Appendix C. However, it is always implied present, and the factor of H is implied extracted by the extractor for this and for the following equalities, method of the extraction is straightforward.

By moving the K component to the left-hand side of the (77) equality, the extractor gets

$$(p_k - v_k) K = -G - \zeta I_k - \chi J_k + p_k (P_{s_k} + \zeta U_{s_k} - \omega A_{s_k}) + v_k (\omega A_k^{\mathbf{tmp}} + \chi \hat{U}_k) \,, \tag{78}$$

that is, expresses K as a linear combination (78) of $G, I_k, J_k, P_{s_k}, U_{s_k}, A_{s_k}, A_k^{\mathbf{tmp}}, \hat{U}_k, H$. However, according to the EFLRSLWB scheme, all these elements are part of the pre-image of K and, hence, K is orthogonal to all of them. Thus, under the supposition $\mathbf{v} \neq \mathbf{p}$ the extractor breaks the DL assumption, which is impossible, so the supposition is incorrect and there holds

$$\mathbf{v} = \mathbf{p} \,. \tag{79}$$

Using the equality (79), the equality (77) rewrites as

$$G + \zeta I_k + \chi J_k = p_k (\, P_{s_k} + \zeta U_{s_k} + \chi \hat{U}_k + \omega (A_k^{\mathbf{tmp}} - A_{s_k}) \,) \,. \tag{80}$$

Note that for the equality (80) the following holds

$$p_k \neq 0 \;\; \text{for each } k \in [0 \dots l-1] \,, \tag{81}$$

since $p_k = 0$ for some k requires that the left-hand side of the equality (80) be equal to zero, however the left-hand side contains non-zero element G alongside with the randomly weighted elements I_k, J_k, and, hence there is only

negligible probability for it to be equal to zero. The implicit presence of H component in the equality (80) does not change the case; if the assertion (81) does not hold then the extractor breaks the DL assumption.

All elements in the right-hand part of the relation (80), namely $P_{s_k}, U_{s_k}, A_k^{\mathbf{tmp}}, A_{s_k}, H$, are in the pre-image of $\hat{U}_k$. Thus, $\hat{U}_k$ is orthogonal to all of them, and hence, due to random weighting by χ to the accuracy of H, the following equality holds

$$G + \zeta I_k = p_k(P_{s_k} + \zeta U_{s_k} + \omega(A_k^{\mathbf{tmp}} - A_{s_k})) \ . \tag{82}$$

In other words, the equality (82) follows from the equality (80) by Theorem 3, where the triplets are taken as

$$(P_{s_k} + \zeta U_{s_k} + \omega(A_k^{\mathbf{tmp}} - A_{s_k}),\ \hat{U}_k,\ 0) \ \text{ and } \ (G + \zeta I_k,\ J_k,\ 0) \ .$$

Suppose that $(A_k^{\mathbf{tmp}} - A_{s_k}) \neq 0$. By unwinding and resuming the $\texttt{zkLin22Choice}_{l,n,l}$ call with different ω' the extractor obtains different p'_k and, subtracting two instances of the equality (82) from each other, obtains

$$0 = p_k(P_{s_k} + \zeta U_{s_k} + \omega(A_k^{\mathbf{tmp}} - A_{s_k})) - p'_k(P_{s_k} + \zeta U_{s_k} + \omega'(A_k^{\mathbf{tmp}} - A_{s_k})) \ ,$$

which rewrites as

$$(p'_k - p_k)(P_{s_k} + \zeta U_{s_k}) = (p_k\omega - p'_k\omega')(A_k^{\mathbf{tmp}} - A_{s_k}) \ . \tag{83}$$

Due to the orthogonality of P_{s_k} and U_{s_k} in the EFLRSLWB scheme, there holds

$$(P_{s_k} + \zeta U_{s_k}) \neq 0.$$

If $p'_k = p_k$ then the left-hand side of the equality (83) is zero, and hence $\omega' = \omega$, that holds only with negligible probability. So, with overwhelming probability $p'_k \neq p_k$ and the extractor divides the equality (83) by $(p'_k - p_k)$, calculating scalar factor a as follows

$$P_{s_k} + \zeta U_{s_k} = a\,(A_k^{\mathbf{tmp}} - A_{s_k}) \ , \ \text{where } a = \frac{p_k\omega - p'_k\omega'}{p'_k - p_k} \ . \tag{84}$$

Unwinding and resuming the $\texttt{zkLin22Choice}_{l,n,l}$ call with different ζ' a couple of times, the extractor calculates factor a' such that

$$P_{s_k} + \zeta' U_{s_k} = a'\,(A_k^{\mathbf{tmp}} - A_{s_k}) \ . \tag{85}$$

Subtracting the equality (84) from the equality (85) and dividing by $(\zeta' - \zeta)$, which is non-zero with overwhelming probability, the extractor obtains

$$U_{s_k} = \frac{a' - a}{\zeta' - \zeta}\,(A_k^{\mathbf{tmp}} - A_{s_k}) \ . \tag{86}$$

Also, it obtains from the equalities (84) and (86)

$$P_{s_k} = \left(a - \zeta\,\frac{a' - a}{\zeta' - \zeta} \right)\,(A_k^{\mathbf{tmp}} - A_{s_k}) \ . \tag{87}$$

After that, as $U_{s_k} \neq 0$, and hence $(a' - a) \neq 0$ in the equality (86), the extractor expresses $(A_k^{\mathbf{tmp}} - A_{s_k})$ through P_{s_k} with it and inserts $(A_k^{\mathbf{tmp}} - A_{s_k})$ into the equality (87), thus obtaining

$$P_{s_k} = \left(a - \zeta\,\frac{a' - a}{\zeta' - \zeta} \right)\,\frac{\zeta' - \zeta}{a' - a}\,U_{s_k} \ . \tag{88}$$

Recalling P_{s_k} and U_{s_k} are orthogonal to each other the extractor breaks the DL assumption with the equality (88), thus the supposition is wrong and there holds

$$A_k^{\mathbf{tmp}} = A_{s_k} \ . \tag{89}$$

In accordance with the equality (89) the equality (82), which is obtained by the extractor after unwinding the $\texttt{zkLin22Choice}_{l,n,l}$ call, rewrites as

$$G + \zeta I_k = p_k(P_{s_k} + \zeta U_{s_k}) \ , \tag{90}$$

where p_k is known to the extractor. Thus the $\texttt{zkLin22Choice}_{l,n,l}$ call is an argument having WEE property for the relation (91).

At the same time, according to the obtained by the extractor equality (89) the same $\texttt{zkLin22Choice}_{l,n,l}$ call is an argument having WEE for the relation (92). Completeness and HVZK of the call follow from Theorem 12. Claim 1 is proven.

O SIGNATURE MULTRATUG FOR L ⩾ 1

Proof: [Theorem 13] We first make the following statement.

Claim 1:
The call to $\texttt{zkLin22Choice}_{l,n,l}$ *in the last step of the EFLRSLWB (Multratug) scheme in Figure 21 is a complete, HVZK argument having WEE for the relation (18) with appropriate input renaming, i.e. for the relation*

$$\mathcal{R} = \left\{ \begin{array}{l} (\mathbf{P} + \zeta\mathbf{U}),\, \mathbf{G}_{[:n]} \in \mathbb{G}^{n*}, H \in \mathbb{G}^*,\, (\{G\}^l + \zeta\mathbf{I}) \in \mathbb{G}^l; \\ \mathbf{s} \in [0 \ldots n-1]^l, \mathbf{p}, \boldsymbol{\alpha} \in \mathbb{F}_{\mathsf{p}}^l \end{array} \middle| \begin{array}{l} \forall k \in [0 \ldots l-1] : \\ G + \zeta I_k = p_k(P_{s_k} + \zeta U_{s_k}) + \alpha_k H \end{array} \right\}, \tag{91}$$

and is also a complete, HVZK argument having WEE for the relation

$$\mathcal{R}' = \left\{ \begin{array}{l} \mathbf{A} \in \mathbb{G}^n,\, \mathbf{A}^{\mathbf{tmp}} \in \mathbb{G}^l,\, H \in \mathbb{G}^*; \\ \mathbf{s} \in [0 \ldots n-1]^l,\, \boldsymbol{\beta} \in \mathbb{F}_{\mathsf{p}}^l \end{array} \middle| \begin{array}{l} \forall k \in [0 \ldots l-1] : \\ A_k^{\mathbf{tmp}} = A_{s_k} + \beta_k H \end{array} \right\}, \tag{92}$$

such that witness $\mathbf{s}$ *is common to the relations (91) and (92).*

Proof: can be found in Appendix N.

Now let's note that the vectors $\mathbf{A}^{\mathbf{tmp}}$ and $\mathbf{J}$ are indistinguishable from white noise, because according to Figure 21 all their elements contain independent blinding components with randomized factors from, respectively, $\boldsymbol{\mu}$ and $\boldsymbol{\upsilon}$.

We have obtained that in the last step of the EFLRSLWB scheme there is a call to the complete, HVZK, and WEE proving system $\texttt{zkLin22Choice}_{l,n,l}$ producing a proof of the relation (91), which is actually the relation (18) with proper renaming. In addition to this, all previous steps of the EFLRSLWB scheme do all the play of the EFLRSL scheme from Figure 14 up to the proof of the relation (18). As for the vectors $\mathbf{A}^{\mathbf{tmp}}$ and $\mathbf{J}$ which are all indistinguishable from white noise, they can be discarded as uninfluential. Thus, we see that the EFLRSLWB scheme is the EFLRSL scheme with the substituted underlying proving system, which is also complete, HVZK, and WEE.

Therefore, the EFLRSLWB scheme is a linkable threshold ring signature with the properties 1...8), which hold due to exactly the same reasons as the properties 1...8) of the EFRLSL scheme in Theorem 9.

The property 9) holds due to the `zk2ElemComm` call in the last step of the EFLRSLWB scheme. By Theorem 1 there holds

$$A^{\mathbf{sum}} = \sum_{k=0}^{l-1} \mathbf{A}_{[k]}^{\mathbf{tmp}} + f_H H + f_D D\ , \tag{93}$$

where f_H, f_D are scalars known to prover. At the same time, by Claim 1 according to the relation (92), the equality (93) unfolds as

$$A^{\mathbf{sum}} = \sum_{k=0}^{l-1} A_{s_k} + \left(f_H + \sum_{k=0}^{l-1} \beta_k \right) H + f_D D\ . \tag{94}$$

Recalling that according to the EFLRSLWB scheme the generator H is an $\mathcal{H}_{\mathbf{point}}$ image of the $A^{\mathbf{sum}}, \mathbf{A}, D$ elements, the equality (94) reduces to

$$A^{\mathbf{sum}} = \sum_{k=0}^{l-1} A_{s_k} + f_D D\ ,$$

which is exactly what the property 9) is. Theorem 13 is proven.

P RANDOM WEIGHTING FOR T-TUPLES

When moving from the equality (52) to the system (53a, 53b, 53c, 53d) we implicitly use Theorem 3. More details about this in Appendix N, where the equality (52) corresponds to the equality (80). However, the transition from the equality (56) to the system (57a, 57c, 57c, 57d, 57e) in Section 6.5 may not seem apparent. To make it clear, we formulate the following Theorem 17, which is a generalization of Theorem 3 to $(t+1)$-element tuples.

Theorem 17:
For any $t \in \mathbb{N}^*$*, for an element vector* $\mathbf{Q} \in \mathbb{G}^t$ *such that* $\mathbf{Q}$ *contains at least one non-zero element, i.e.* $\mathrm{nz}(\mathbf{Q}) \neq \varnothing$*, for two non-zero elements* $P, H \in \mathbb{G}^*$ *such that* $P\,!= \mathrm{lin}(\mathrm{nz}(\mathbf{Q}) \cup H)$ *and* $H\,!= \mathrm{lin}(\mathrm{nz}(\mathbf{Q}) \cup P)$ *hold, the protocol* `zkTElemRW` *in Figure 27 is a complete, HVZK argument having WEE for the relation (95).*

$$\mathcal{R} = \left\{ \begin{array}{l} P \in \mathbb{G}^*,\ \mathbf{Q} \in \mathbb{G}^t,\ H \in \mathbb{G}^*,\ Z \in \mathbb{G},\ \mathbf{F} \in \mathbb{G}^t; \\ a, \alpha \in \mathbb{F}_{\bar{\mathfrak{p}}},\ \boldsymbol{\beta} \in \mathbb{F}_{\bar{\mathfrak{p}}}^t \end{array} \middle| \begin{array}{l} Z = aP + \alpha H \ \wedge \\ \mathbf{F} = a\mathbf{Q} + \boldsymbol{\beta} H \end{array} \right\}, \tag{95}$$

$\boxed{\texttt{zkTElemRW}(P, \mathbf{Q}, H, Z, \mathbf{F};\ a, \alpha, \boldsymbol{\beta})}$

Relation $\mathcal{R} = \left\{ \begin{array}{l} P \in \mathbb{G}^*,\ \mathbf{Q} \in \mathbb{G}^t,\ H \in \mathbb{G}^*,\ Z \in \mathbb{G},\ \mathbf{F} \in \mathbb{G}^t; \\ a, \alpha \in \mathbb{F}_{\bar{\mathfrak{p}}},\ \boldsymbol{\beta} \in \mathbb{F}_{\bar{\mathfrak{p}}}^t \end{array} \middle| \begin{array}{l} Z = aP + \alpha H \ \wedge \\ \mathbf{F} = a\mathbf{Q} + \boldsymbol{\beta} H \end{array} \right\}$ // (95)

// $P, \mathbf{Q}, H$ in $\mathcal{R}$ satisfy $\mathrm{nz}(\mathbf{Q}) \neq \varnothing \ \wedge\ P\ != \mathrm{lin}(\mathrm{nz}(\mathbf{Q}) \cup H) \ \wedge\ H\ != \mathrm{lin}(\mathrm{nz}(\mathbf{Q}) \cup P)$

$\mathcal{P}$'s input : $(P, \mathbf{Q}, H, Z, \mathbf{F};\ a, \alpha, \boldsymbol{\beta}))$

$\mathcal{V}$'s input : $(P, \mathbf{Q}, H, Z, \mathbf{F})$

$\mathcal{P}$'s output : none

$\mathcal{V}$'s output: $Accept$ or $Reject$

$\boxed{\mathcal{V}}$: $\boldsymbol{\delta} \leftarrow\!\$\ \mathbb{F}_{\bar{\mathfrak{p}}}^{t*}$

$\boxed{\mathcal{V} \rightarrow \mathcal{P}}$: $\boldsymbol{\delta}$

$\boxed{\mathcal{P}}$: computes $\hat{\alpha} = \alpha + \langle \boldsymbol{\delta}, \boldsymbol{\beta} \rangle$

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: compute $X = P + \langle \boldsymbol{\delta}, \mathbf{Q} \rangle$

$Y = Z + \langle \boldsymbol{\delta}, \mathbf{F} \rangle$

and run any complete, HVZK, and WEE protocol that convinces $\mathcal{V}$ that

$a, \alpha, \boldsymbol{\beta}$ at $\mathcal{P}$'s privare input connect X and Y so that $Y = aX + \hat{\alpha} H$

Figure 27: Zero-knowledge argument for two t-tuples proportional to each other

Proof: (Informal.) The words about completeness, HVZK, and H from Appendix C apply here as well, so we exclude the H components from the consideration. Proving WEE as follows.

Witness extractor extracts a and thus obtains the equality

$$Z + \langle \boldsymbol{\delta}, \mathbf{F} \rangle = aP + a \langle \boldsymbol{\delta}, \mathbf{Q} \rangle \ . \tag{96}$$

Unwinding and repeating the procedure for different challenge vector $\boldsymbol{\delta}'$ the extractor obtains a' such that

$$Z + \langle \boldsymbol{\delta}', \mathbf{F} \rangle = a'P + a' \langle \boldsymbol{\delta}', \mathbf{Q} \rangle \ . \tag{97}$$

Eliminating Z from the equalities (96) and (97), the extractor gets

$$\langle \boldsymbol{\delta}' - \boldsymbol{\delta},\ \mathbf{F} \rangle = (a' - a)P + \langle a'\boldsymbol{\delta}' - a\boldsymbol{\delta},\ \mathbf{Q} \rangle \ . \tag{98}$$

Suppose, $a \neq a'$, then the extractor eliminates the first non-zero $F \in \mathbf{F}$ in the equality (98) exactly the same way as it did with Z. Repeating the process it completely eliminates $\mathbf{F}$, thus discovering that

$$P = \mathrm{lin}(\mathrm{nz}(\mathbf{Q})) \ ,$$

which contradicts to the premise $P\ != \mathrm{lin}(\mathrm{nz}(\mathbf{Q}))$. Therefore, the supposition is incorrect and with overwhelming probability there holds $a = a'$ for any pair of random challenge vectors $\boldsymbol{\delta}$ and $\boldsymbol{\delta}'$. This turns the equality (98) into the equality

$$\langle \boldsymbol{\delta}' - \boldsymbol{\delta},\ \mathbf{F} - a\mathbf{Q} \rangle = 0 \ ,$$

which is the classical random weighting for the arbitrary elements $\mathbf{F} - a\mathbf{Q}$, that implies $\mathbf{F} - a\mathbf{Q} = \{0\}^t$. Thus, Theorem 17 is proven.

Returning to the equality (56), in which the element $\hat{U}_k$ is orthogonal to all other elements in the right-hand side, i.e. where holds $\hat{U}_k\ != \mathrm{lin}(H, P_{s_k}, U_{s_k}, A_{s_k}, A_k^{\mathbf{tmp}}, \mathbf{U}_k^{\mathbf{tmp}})$, and where P_{s_k} is guaranteed non-zero, the system (57a, 57c, 57c, 57d, 57e) immediately follows from it by Theorem 17.

Q NOTES ABOUT DUALRING-EC

We thank Tsz Hon Yuen et al. for the work [15] that led us to the optimization idea in Section 6.4. However, the DualRing-EC signature, according to the paper [15], requires all keys in the ring to be honestly generated, i.e. it doesn't work with malformed ones. In contrast, our security model defined by Theorem 9 doesn't depricate malformed keys in the rings. We have tried to assess whether an environment in which EFLRSL remains secure can be used for DualRing-EC, and discovered the following attack, of course, with reference to our security model.

Let a dishonest $\mathcal{P}$ want to sign with DualRing-EC using a ring of four malformed public keys, none of which it knows a secret key for. Knowing no secret keys for Q, R, K and knowing a secret key for P, it creates the four-element ring as $\{Q, R, P + K, P - K\}$. Then $\mathcal{P}$ performs as though it signs honestly with P's secret key using three-element ring $\{Q, R, P\}$. However, it still hashes the four-element ring to create the challenge. Instead of creating the Sum Argument [15] for three challenges c_0, c_1, c_2, which correspond to Q, R, P, it splits c_2 into two halves and includes the Sum Argument for four challenges $c_0, c_1, c_2/2, c_2/2$ into the forgery. After that, honest $\mathcal{V}$ accepts this signature.

R LOW ANONYMITY OF U/X

Let's determine anonymity implications of having in a public transcript an element of the form $x^{-1}U$ such that U is a fixed generator and x is a private key. It may not necessarily be a linking tag, such element may appear, for instance, in a part of the scheme proving the balance.

Consider a rather simple and therefore very possible case of non-uniform distribution of x's. Let the distribution have a high probability for pairs of private keys (x_1, x_2) such that $x_2 = 2x_1$. Consequently, two signatures signed with keys from the same pair will be linked together by checking whether the element $x_2^{-1}U$ multiplyed by 2 is equal to its counterpart.

The obvious objection to this case is that the system may by design forbid such tightly coupled keys. This is, for example, the case in [13], where private keys behind the public keys in the rings have the form $x = b + r$ with hidden b, and independently and uniformly distributed r which may be seen to an adversary. Thus, the element in question takes the form

$$(b + r)^{-1}U\ , \quad \text{where } r \text{ is known to the adversary, and always is independently and uniformly distributed.}$$

According to [6, 14, 9], this form makes it impossible to break anonymity, even if the adversary is diligently observing r.

Takeaway from this is that if a scheme conatains an element of the form $x^{-1}U$, then it is not anonymous w.r.t. chosen public key attackers. Also, in this case it is impossible to prove with the usual methods, for example, existential unforgeability against adaptive chosen message / public key attackers, even if the scheme possesses this property.